FOR DISCUSSION ONLY

add'l number

WORKINGPAPER

NRE ECONOMIC RESEARCH SERVICE

UNITED STATES DEPARTMENT OF AGRICULTURE

24510

Spatial Dimensions of Land Use and Environmental Change Using the Conservation Needs Inventory: Methodology and Applications

by

Ralph E. Heimlich
and
Clayton W. Ogg

August 1978

United States Department of Agriculture
Economics, Statistics, and Cooperatives Service
Natural Resource Economics Division
Northeast Resource Program Group
1974 Sproul Road, 4th Floor
Broomall, PA 19008 (suburban Philadelphia)
Phone: (215) 353-6889

Abstract

Two computer-based methodologies which make use of the spatial information contained in the Conservation Needs Inventory (CNI) are presented and illustrated with applications to the Chowan-Pasquotank River Basin. The amount and location of land clearing and the amount and location of gross soil erosion from cropland are determined and displayed on computer-generated maps. Potential applications to river basin planning and other studies are discussed.

Keywords: CNI, computer mapping, soil erosion, soil mapping, LP

TABLE OF CONTENTS

TABLES

FIGURES

Introduction

The Conservation Needs Inventory (CNI) of 1967 is an important data base for soil, land use and treatment needs information which is consistent throughout the United States. In the past, however, CNI information has been used in an aggregated form, primarily as tables of land use and soils by county or region. While the sampling method employed by the CNI is only valid for individual counties or larger geographic areas, much of the spatial information contained in the data has not been used.

This paper explains two methodologies which make fuller use of the spatial dimension inherent in the individual point samples of the CNI data base: 1) computer mapping based on the CNI primary sample units (PSUs); and 2) computation of gross soil erosion using the Universal Soil Loss equation evaluated at CNI sample points. Applications to the Chowan-Pasquotank River basin of Virginia and North Carolina are presented. Listings of the computer programs developed are included in appendices.

Such techniques should increase the usefulness of the existing CNI data base as well as the 1977 update. Computer methods used in conjunction with this data base can result in more thorough, varied analysis of river basin, regional and other planning issues. Such methods are likely to be more economical as well.

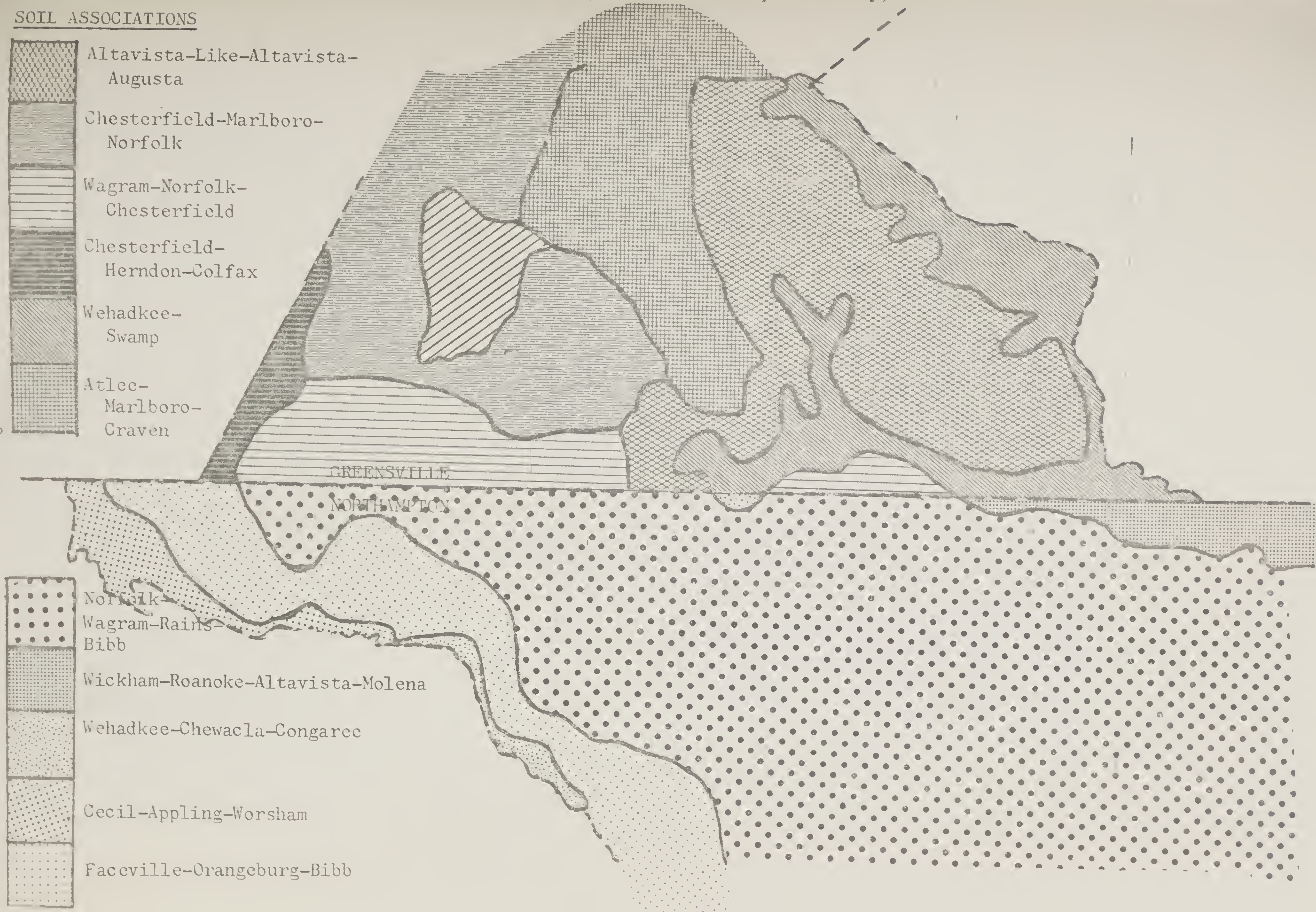
Planning Limitations of Existing Soil Maps

Mapped soils information is generally available to planners in two forms: 1) as detailed soil maps showing the location of each soil on a relatively large scale (small area) map; or 2) as soil association maps showing the general location of broad groups of soils. Detailed soil maps are often not useful for planning efforts at the river basin level because they provide an overabundance of information. Soil association maps, on the other hand, obscure important distinctions relevant to large area planning and are often not consistent across political boundaries.

Soil associations are logical groupings of soils with similar parent material that are found physically to occur together. Although associated geographically, the soils in an association may differ in their suitability for agricultural production. For example, the Norfolk-Wagram-Rains-Bibb soil association in Northampton County, NC, shown in Figure 1, consists of soils which are good (60%), good to fair (20%), fair (10%), and poor (10%) for agricultural use.

A second difficulty with some of the more detailed soil association maps is that they were originally prepared for Soil and Water Conservation Districts, which approximate county boundaries. Different districts group soils into different associations, making cross-county comparisons in larger planning areas difficult. The problem is compounded in multi-state planning areas, as shown in Figure 1, because not only are associations composed of different soils, but similar soils are often given different names in different states. These problems can be overcome only through considerable time and effort on the part of soils experts.

Figure 1-- SOIL ASSOCIATIONS
Greensville County, VA and Northampton County, NC



Detailed soil maps avoid the difficulties encountered in using soil associations because the productivity characteristics of individual soils are known and because they are mapped as discrete units. An overaboundance of information makes detailed soil maps less useful for planning purposes. For example, there are 18 different soils mapped in just one square mile of Figure 2. Only about 55% of the nation has been mapped, so many areas across the country do not have detailed soil maps. Even when available, detailed soil maps must be generalized to a more useable level of detail. Also, river basin planning studies produce large amounts of information which cannot be effectively overlaid with detailed soil maps without rescaling. This paper will argue that computerized soils data currently available can be used much more efficiently than detailed soil maps that would first have to be digitized or handled through cartographic centers for generalization and rescaling.

The Conservation Needs Inventory: A Primer

In order to more fully appreciate the kind of information available from the detailed sample records collected during the 1967 Conservation Needs Inventory (CNI), a brief review of the methodology and terminology used is presented.

Stratified, random sample areas were selected by the Statistical Laboratory at Ames, Iowa, by dividing each county in the nation into strata with 48 sample areas in each. One sample area was selected at random from each stratum. Sample areas varied from 40 to 640 acres each and were mapped on county base maps supplied by the State SCS. These sample areas are called Primary Sample Units (PSU) and represent approximately two percent of the area of each county. (Soils Memorandum 21(revision 2), November 10, 1965).

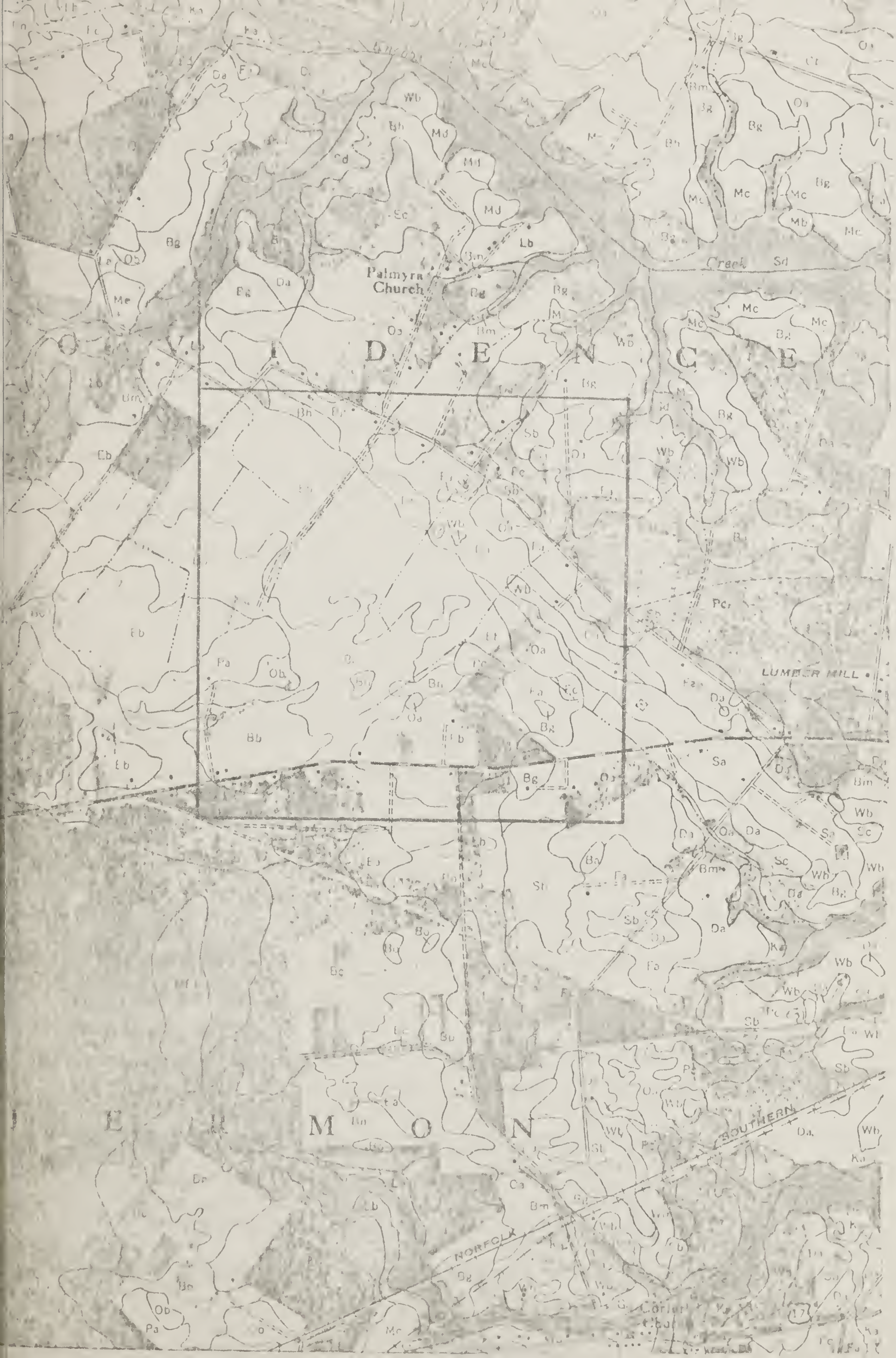


Figure 2--Detailed Soil Map, Northampton County, NC

The PSUs were superimposed on soil maps of the county and a sampling template oriented randomly over the PSU to obtain several dozen individual sample points. Soil information for each of the individual sample points was read directly from the soil map and land use and treatment information was obtained through field checks. (Conservation Needs Memorandum 21, November 3, 1965).

The Statistical Laboratory expanded the PSU data to inventory acreages of land capability classes and subclasses obtained during the same year, 1967. Data was reviewed and adjusted by state and county CNI committees organized by the State SCS.

Magnetic tapes were prepared for each state containing one record of 24 fields for each individual sample point. Each record has state, county, soil conservation district, watershed and PSU identifying codes; soil, slope, and land capability codes; land use and conservation treatment need codes; and an acreage showing the area of the county represented by the point. There are 149,459 records for North Carolina and 97,980 for Virginia, with 36,505 records in the 26 counties of the Chowan Basin.

A subsample of the 1967 CNI was resampled in 1977. Two to three points for selected PSUs were sampled obtaining updated land use and treatment needs, data necessary to evaluate the Universal Soil Loss Equation (USLE) and other data.

An Alternative Method for Mapping Soils Information

Soils data for broad planning purposes can be mapped relatively easily using computer based data. Soil mapping units thus derived are consistent across county and state boundaries, like detailed soil maps, but soils are grouped for planning purposes. The process entails: 1) grouping soils into units meaningful for the particular study; 2) sampling the location

of these units in space; and 3) mapping the sampled locations and their associated values. A description of the three parts of the process follows.

Establishing Meaningful Soil Groups - The first step in the process is to group individual soils into fewer units which will be easier to manipulate and which will have characteristics meaningful to the study. For the Chowan River Basin, agricultural productivity was of primary interest, hence soil productivity groups (SPG's) were identified and created. The soils of North Carolina were arranged into 32 groups based on crop yield characteristics, management practices, and land treatment needs for the Tar-Neuse River Basin study using CNI data. (Parks, 1975). Because Virginia soils names differ slightly from those used for the same soils in North Carolina, the help of Virginia soil scientist, Richard Googins, was enlisted to extend the North Carolina SPG's to Virginia soils. Thus, a uniform, consistent classification of soils based on productivity and conservation needs was established transcending county and state boundaries.

Locating Sample Units - In order to map SPG's it is necessary to determine their location in space at a large number of points. For this purpose the CNI sample record provided an adequate sample of the location of SPGs.

Commensurate with the level of detail necessary in the data and in order to reduce the required effort, the 36,505 CNI records within the Chowan Basin were aggregated to the PSU level by selecting the major SPG, land use and treatment need, on the basis of the acreage represented on the individual records making up the PSU. The process, diagrammed in Figure 3, resulted in 1,408 aggregated records, one record for each PSU,

Figure 3
Processing Primary Sample Unit (PSU) Records

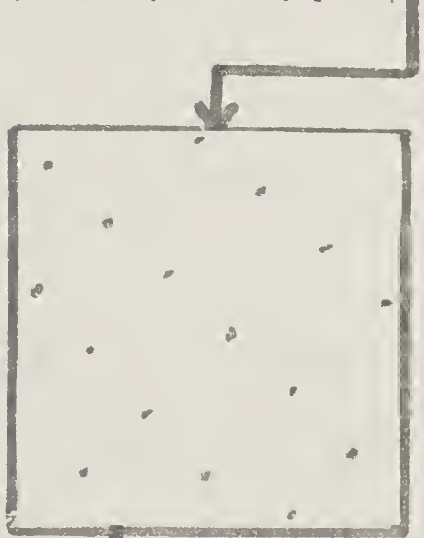
County Base Map

locates PSUs

Primary Sampling Unit

with individual

sample points



Individual Sample Records

```
330087680502O1R31133 P14021200    13        20043- - 000 3W1  F14-00201986  *CHEWACLA SIL
33008768050201R31133 P14021200    13        19043- - 000 3W1  F14-00201986  *CHEWACLA SIL
33008768050201R31133 P14021200    13        18043- - 000 3W1  F14-00201986  *CHEWACLA SIL
33008768050201R31133 P14021200    13        17014- - 000 2W1  F14-00201986  *CONGAREE SIL
33008768050201R31133 P14021200    13        16043- - 000 3W1  F14-00201986  *CHEWACLA SIL
33008768050201R31133 P14021200    13        15043- - 000 3W1  F14-00201986  *CHEWACLA SIL
```

Aggregated PSU Record

```
33008020402R 153 T14021200 07271496 26 944 F1000 F-1000 35525 77010
33008020501R 153 T14021200 07473482 20 432 L 729 L2 486 35535 76575
33008020503R 153 T14021200 06867524 261000 F1000 F-1000 35505 76535
33008020601R 153 T14021200 06463552 261000 F1000 F-1000 35550 76465
```

for a 96% reduction in data from the original 36,505 sample point records.

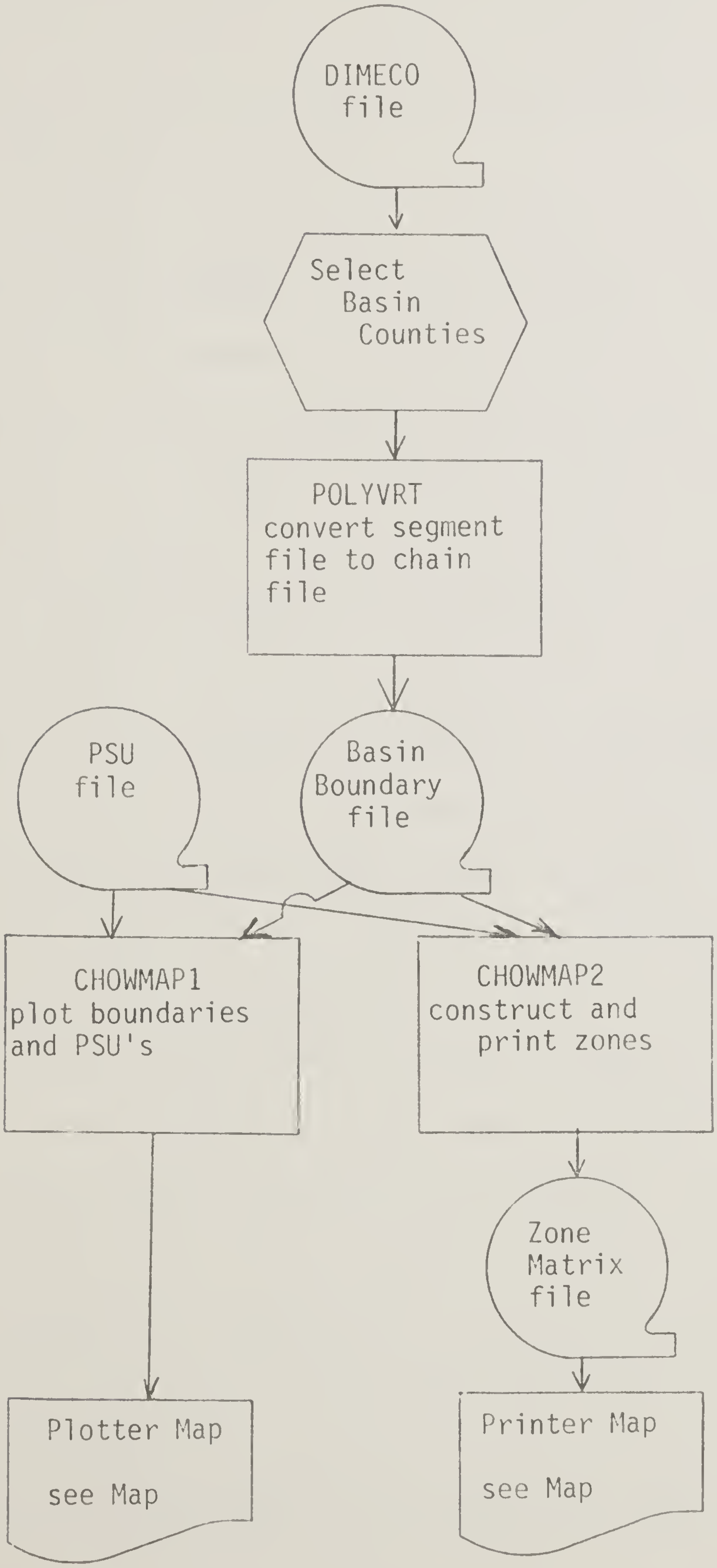
Since CNI records contain no geographical coordinates, microfilm copies of the county base maps, with PSU's indicated, were obtained and large paper copies produced on which latitude and longitude lines were superimposed. The coordinates of each PSU were determined to the nearest half minute and were appended to the aggregated CNI record for each PSU. The resulting data base consisted of 1,408 records, one for each PSU, with the dominant SPG, 1967 land use and 1967 treatment need, and geographic coordinates. Including geographic coordinates among the data contained in each sample point record would greatly improve the CNI and any such future data gathering efforts.

Mapping Sample Units - In order to graphically display the data, coordinates on the aggregated PSU records were combined with county boundaries contained on the Census Bureau's Dual Independent Map Encoding County File (DIMECO) as shown in Figure 4.

First, the county boundaries of interest in the Chowan Basin were selected from the DIMECO file, which contains boundaries for all states and counties. Next, the segment file structure of DIMECO was converted into a different file structure in which "chains" of segments making up entire polygons were strung together from the individual boundary segments. This is an important step in that most mapping program packages, such as SYMAP and CALFORM, use a form of this chain file structure in their input data. Conversion of the Basin boundary file was accomplished using POLYVRT, a program package produced by the Harvard Laboratory for Computer Graphics and Spatial Analysis. (POLYVRT User's Manual, November, 1974)

Two kinds of graphical display were produced from the data contained in the aggregated PSU records and in the Basin boundary file: 1) plotter

Figure 4
Mapping of Sample Units



maps showing the boundary of counties in the Basin and the location and identity of each PSU, as in Figure 5; and 2) printer maps showing shaded zones based on the PSU's as in Figure 9. The programs which accomplish this mapping are briefly described below and are discussed in more detail in Appendices A-1 and A-2.

Plotter mapping is the simpler of the two forms, and is accomplished by a program called CHOWMAP1 which was written for this project. The program scales the geographic coordinates of boundary points and data points, converts them to a Cartesian coordinate system used by the plotter, limits the plot to a "window" specified by control cards through a process called "clipping," and controls the movements of the plotter pen to draw the county boundaries and indicate the location and identification of each data point. Data points consisted of the PSU's, which could be identified by the associated SPG, land use, or treatment need, or by any other identification field in the aggregated PSU record.

Printer mapping is more complicated. As shown in Figure 6, the process begins with the initial map matrix on which land boundaries and the SPG identifications for each PSU are located. For each empty cell in the map matrix which is inside the land boundaries, an SPG is assigned based on the SPG of the nearest PSU. When all empty cells within the boundary are assigned, the map matrix is complete. The computer program actually works with a numerical representation of the map matrix called a scandeck (See Appendix A-2). Using the computer mapping package SCANGEN (Center for Census Use Studies, May, 1975) any vector of data associated with SPG's can be mapped into the zones defined by the scandeck. Thus, zones are created only once and stored for use in mapping various

Figure 5--Plotter map of PSU's identified by Soil Productivity Group (SPG)
Albemarle Sound, North Carolina

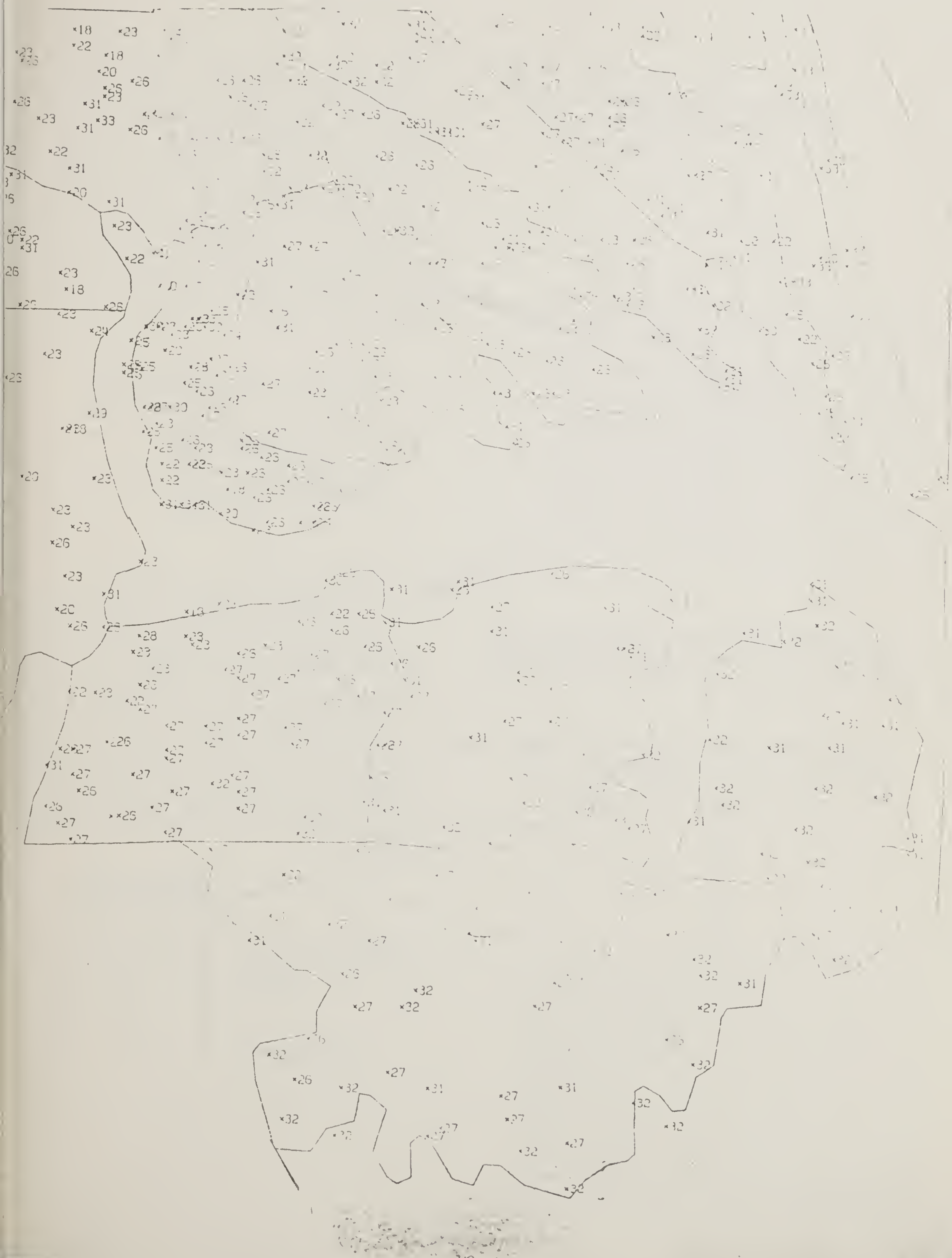
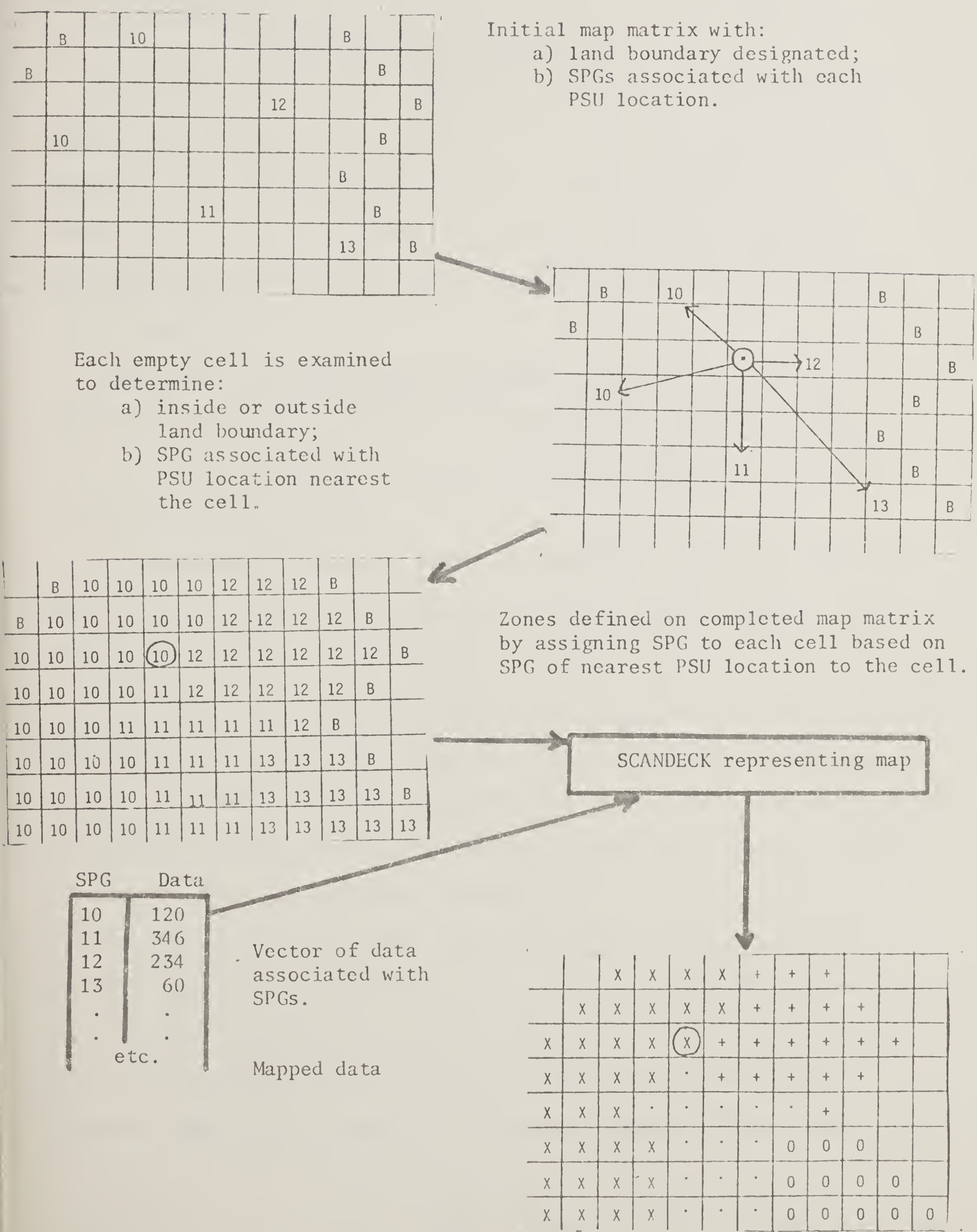


Figure 6 -- Creation of Scandeck and Printer Maps



data associated with them. This is a distinct advantage over similar mapping packages which must create zones each time a map is desired.

Application to the Chowan Basin

Two applications of this methodology to river basin planning in the Chowan-Pasquotank Basin will be shown to illustrate the utility of the approach. First, the amount and location of land use change around Albemarle Sound was investigated. Second, the amount and location of gross soil erosion from agricultural activities in the Basin was calculated and mapped.

Amount and Location of Land Use Change - Establishing which Soil Productivity Groups are experiencing rapid change to agricultural uses, and which land uses are most affected, indicates the potential for future change through land clearing.

The aggregated PSU records contain the dominant land use, in terms of area, of the PSU from the 1967 Conservation Needs Inventory (CNI). Rapid clearing and draining of forested wetlands made CNI land use data obsolete. In order to update the land use information for wetland soils two kinds of more recent land use data were obtained. The United States Geological Survey publishes rectified 7.5 minute orthophoto quadrangle maps at a scale of 1:24,000 which display adequate shading and texture to distinguish between major land uses wuch as cropland, pasture, forest, and urban areas. In addition, for more limited areas, USGS has prepared Land Use and Land Cover maps from airphoto and satellite data at a scale of 1:250,000. (Anderson, et al., 1976) These maps distinguish between 37 different land uses and cover types, shown in Table 1, and are thus more detailed than the orthophoto maps. Land use and land cover maps showing 1974 land use

were available from USGS for Rocky Mount and Manteo quadrangles. Orthophoto maps showing 1974 land use were obtained for most of the rest of the region surrounding Albemarle Sound from the North Carolina Department of Natural and Economic Resources, as shown in Figure 7.

Land use change between 1967 and 1974 was assessed using the orthophoto quad maps in the Northern Albemarle Sound area by first superimposing a ½-minute grid over each quadrangle and locating each PSU by means of the coordinates on each aggregated PSU record. The 1974 land use was recorded as one of four possibilities: 1) cropland and pasture; 2) forest land; 3) urban land; or 4) water. Each PSU record thus processed contained a land use observation for two points in time. Since the land use and land cover maps were available as translucent overlays, a plot map of the county outlines and aggregated PSU data points for the Southern Albemarle Sound area was created at the same scale as the land use overlay using CHOWMAP 1 and the 1974 land use read directly from the combined map.

The 1974 land use data from both sources was appended to the aggregated PSU records so that 1967 land use, 1974 land use, or the change in land use from 1967 to 1974 could be determined and mapped using CHOWMAP 2. This data could also be displayed in tabular form, as in Table 2, by direct processing of the aggregated PSU records.

<u>Amount and Location of Gross Soil Erosion</u> - Data from individual CNI sample records was used in conjunction with the Universal Soil Loss Equation (USLE) to project 1990 gross soil erosion by Soil Productivity Group in the Chowan-Pasquotank Basin. Erosion was projected under different scenarios as to future production levels in the Basin, and

Table 1--Land use and land cover classifications

LAND USE AND LAND COVER MAPS These maps show the following categories.

LEVEL I	LEVEL II
1 Urban or Built-up Land	11 Residential
	12 Commercial and Services
	13 Industrial
	14 Transportation, Communications, and Utilities
	15 Industrial and Commercial Complexes
	16 Mixed Urban or Built-up Land
	17 Other Urban or Built-up Land
2 Agricultural Land	21 Cropland and Pasture
	22 Orchards, Groves, Vineyards, Nurseries and Ornamental Horticultural Areas
	23 Confined Feeding Operations
	24 Other Agricultural Land
3 Rangeland	31 Herbaceous Rangeland
	32 Shrub and Brush Rangeland
	33 Mixed Rangeland
4 Forest Land	41 Deciduous Forest Land
	42 Evergreen Forest Land
	43 Mixed Forest Land
5 Water	51 Streams and Canals
	52 Lakes
	53 Reservoirs
	54 Bays and Estuaries
6 Wetland	61 Forested Wetland
	62 Nonforested Wetland
7 Barren Land	71 Dry Salt Flats
	72 Beaches
	73 Sandy Areas other than Beaches
	74 Bare Exposed Rock
	75 Strip Mines, Quarries, and Gravel Pits
	76 Transitional Areas
	77 Mixed Barren Land
8 Tundra	81 Shrub and Brush Tundra
	82 Herbaceous Tundra
	83 Bare Ground Tundra
	84 Wet Tundra
	85 Mixed Tundra
9 Perennial Snow or Ice	91 Perennial Snowfields
	92 Glaciers

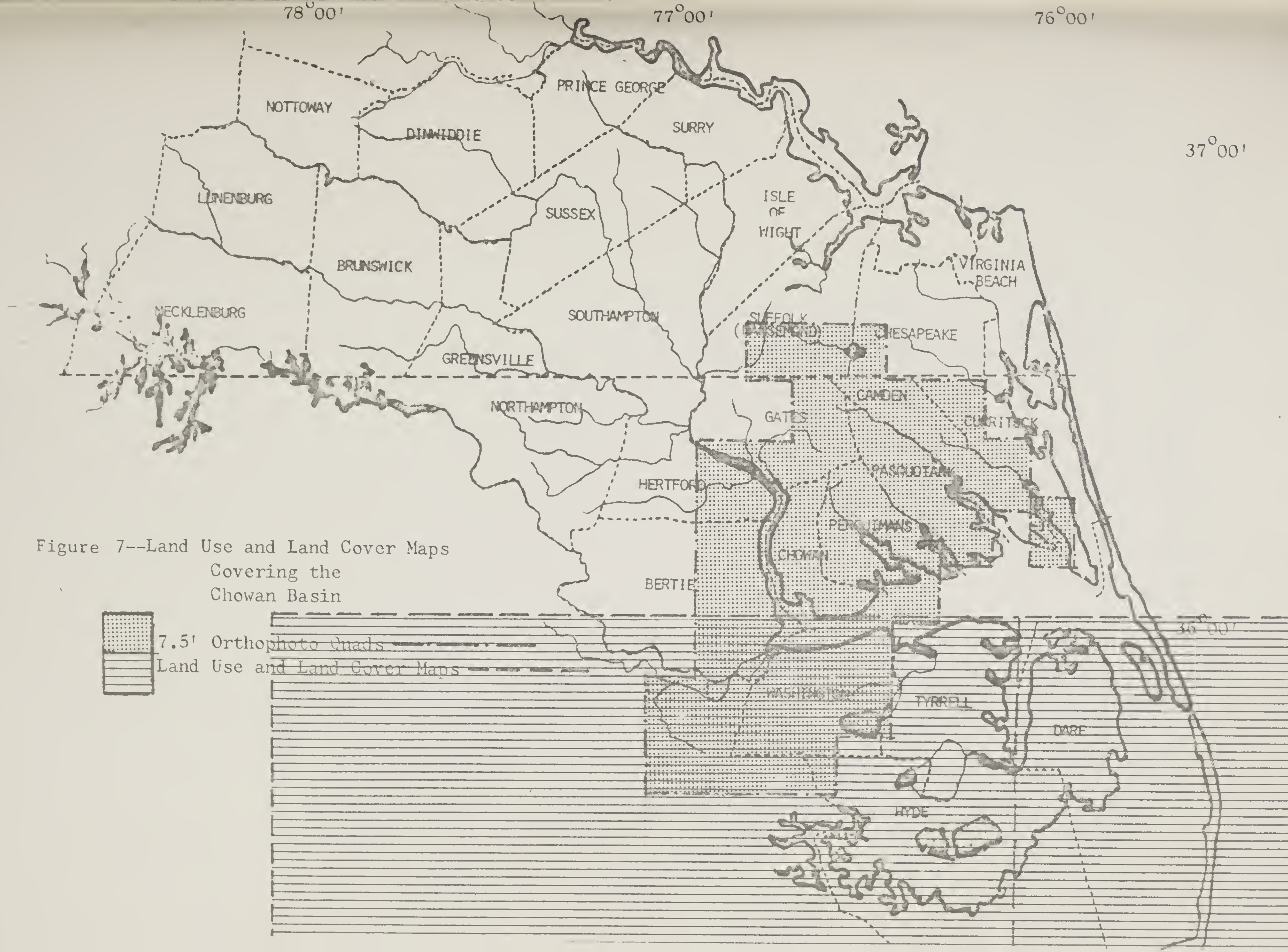


Figure 7--Land Use and Land Cover Maps Covering the Chowan Basin

Table 2-- Soil Productivity Groups with Land Use Changes, Northern Albemarle Sound 1/
1967-1974

PG	1967 Cropland & Pasture	+	New Cropland & Pasture Uses	-	New Forest Uses	=	1974 Cropland & Pasture	1967-1974 Net Change
					acres			
2	22,628		0		4,056		18,572	-4,056
3	60,303		23,340		18,423		65,220	4,917
5	38,518		28,304		11,248		55,574	17,056
6	133,476		39,720		22,521		150,675	17,199
7	23,891		24,833		6,724		42,000	18,109
1	0		1,292		0		1,292	1,292
2	107		11,005		0		11,112	11,005
)TAL	278,923		128,494		62,972		344,445	65,522

1/ Camden, Chowan, Currituck, Gates, Pasquotank, and Perquimans Counties.

economic effects of implementing certain soil conservation plans were compared. These effects include changes in the location of production, based on a linear programming model which was used to anticipate farmers' efforts to find the least costly way of meeting environmental objectives.

The Universal Soil Loss Equation multiplicatively combines a number of factors relevant to the detachment and removal of soil particles on agricultural land. (Wischmeier and Smith, 1965) The equation is:

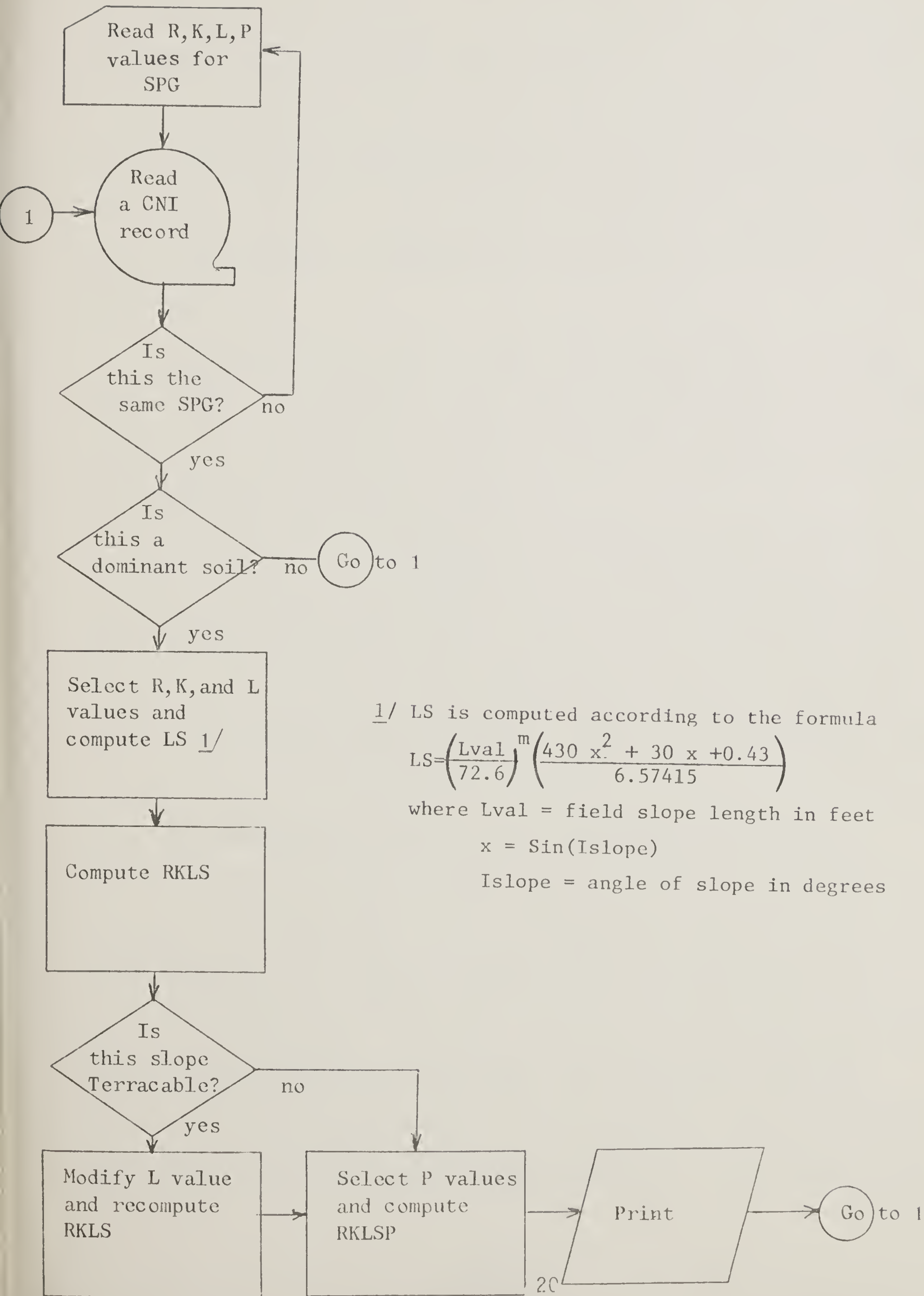
$$A = R \times K \times L \times S \times C \times P$$

where the factors are defined as follows.

The computed soil loss per acre, A, is an estimate of the gross amount of soil which is detached from the land surface over the course of a crop year. No account is made for deposition of sediment on other parts of the same field, so it does not measure the escape of sediment from the field.

The rainfall factor, R, is derived from a map of iso-erodent values measured empirically. The R factor, like many of the USLE factors, is tabulated in the North Carolina Soil Conservation Service Technical Guide. (NCSCS, October, 1976) The computer program which calculates soil loss, diagrammed in Figure 8, selects the R factor value based on the county code from each CNI sample record from a list of R factors tabulated by county, taken from the SCS Guide. The soil erodability factor, K, is tabulated by soil type in the SCS Guide. Examination of soils data for SPG's in the Chowan Basin showed that for each SPG fewer than four soils always accounted for more than two-thirds of the SPG area. The EROSION (Appendix A-3) program examines the soil code on each individual CNI sample record, determines if it is a dominant soil, and assigns the correct K factor value.

Figure 8--Soil loss computation flow chart



1/ LS is computed according to the formula

$$LS=\left(\frac{Lval}{72.6}\right)^{m}\left(\frac{430\ x^{2} + 30\ x + 0.43}{6.57415}\right)$$

where Lval = field slope length in feet

x = Sin(Islope)

Islope = angle of slope in degrees

The slope length factor, L, is perhaps the most difficult factor to evaluate given present data. A survey conducted in 1972 by the Soil Conservation Service determined the dominant soil, slope length, slope, K value and T value for land resource areas in selected states. (McElroy, et al., May 1976, and correspondence with Roy M. Gray, SCS dated 4 April 1977) Using this data a slope length was related to each dominant soil. Thus, a slope length is automatically assigned when the K value is determined.

The slope gradient factor, S, was taken directly from the slope field of the individual CNI sample record. The slope gradient and slope length values are used in the equation for the LS factor contained in SCS (September, 1977, shown in Figure 8.

The product RKLS is computed from the factors listed above and constitutes the basic soil erosion value associated with a particular sample point in a particular SPG, listed in Table 3. The RKLS values for each sample point in an SPG are averaged, using the acreages represented by each sample point as weights to obtain an area-weighted average bare ground soil loss for the SPG. This value is later multiplied by the cropping management factor, C, for the appropriate crops and rotations and by the erosion control practice factor, P, which are tabulated in the SCS Guide. Thus, an estimate of gross soil erosion for the particular soils, slopes, cropping practices, and conservation practices is calculated. Plans for conserving soil and preventing nonpoint pollution primarily analyze the economic and locational impacts of practices with low P and C values on crops and soils with erosion potential. The results of one such plan are shown in Figure 9 for the Chowan Basin.

Table--3 Gross soil erosion, Chowan Basin 1/

SPG.	No Practice	Contour Farming	Field SC A 2/	Field SC B 3/	Contour SC A 4/	Contour SC B 5/
	-----	-----	-----	tons/acre/year	-----	-----
10.	61.2	30.6	21.4	24.5	15.3	21.4
11.	95.6	57.4	38.2	43.0	28.7	38.2
12.	75.4	37.7	26.4	30.2	18.9	26.4
13.	29.2	17.5	10.2	13.1	8.7	11.7
14.	98.1	49.1	34.3	39.2	24.5	34.3
15.	18.5	11.1	6.5	8.3	5.6	7.4
16.	138.9	83.4	55.6	62.5	41.7	55.6
17.	131.1	104.9	65.6	85.2	52.5	65.6
18.	41.2	22.1	14.4	17.2	11.0	15.1
19.	106.4	53.7	37.5	42.8	26.9	37.5
20.	127.2	63.6	44.5	50.9	31.8	44.5
21.	0.0	0.0	0.0	0.0	0.0	0.0
22.	31.1	17.5	10.9	13.4	8.7	11.8
23.	41.1	24.7	14.4	18.5	12.4	16.5
24.	29.1	17.5	10.2	13.1	8.7	11.7
25.	20.5	12.3	7.2	9.2	6.1	8.2
26.	37.3	22.4	13.0	16.7	11.2	14.9
27.	47.0	28.2	16.4	21.1	14.1	18.8
28.	20.8	12.4	7.2	9.4	6.3	8.3
29.	167.0	94.0	63.7	75.5	47.0	63.7
30.	42.6	22.1	14.9	17.4	11.0	15.3
31.	1.1	0.7	0.4	0.5	0.3	0.5

1/ Soil losses shown do not include the effects of the crop practice factor, C, and are thus equivalent to bare-ground or fallow soil lost.

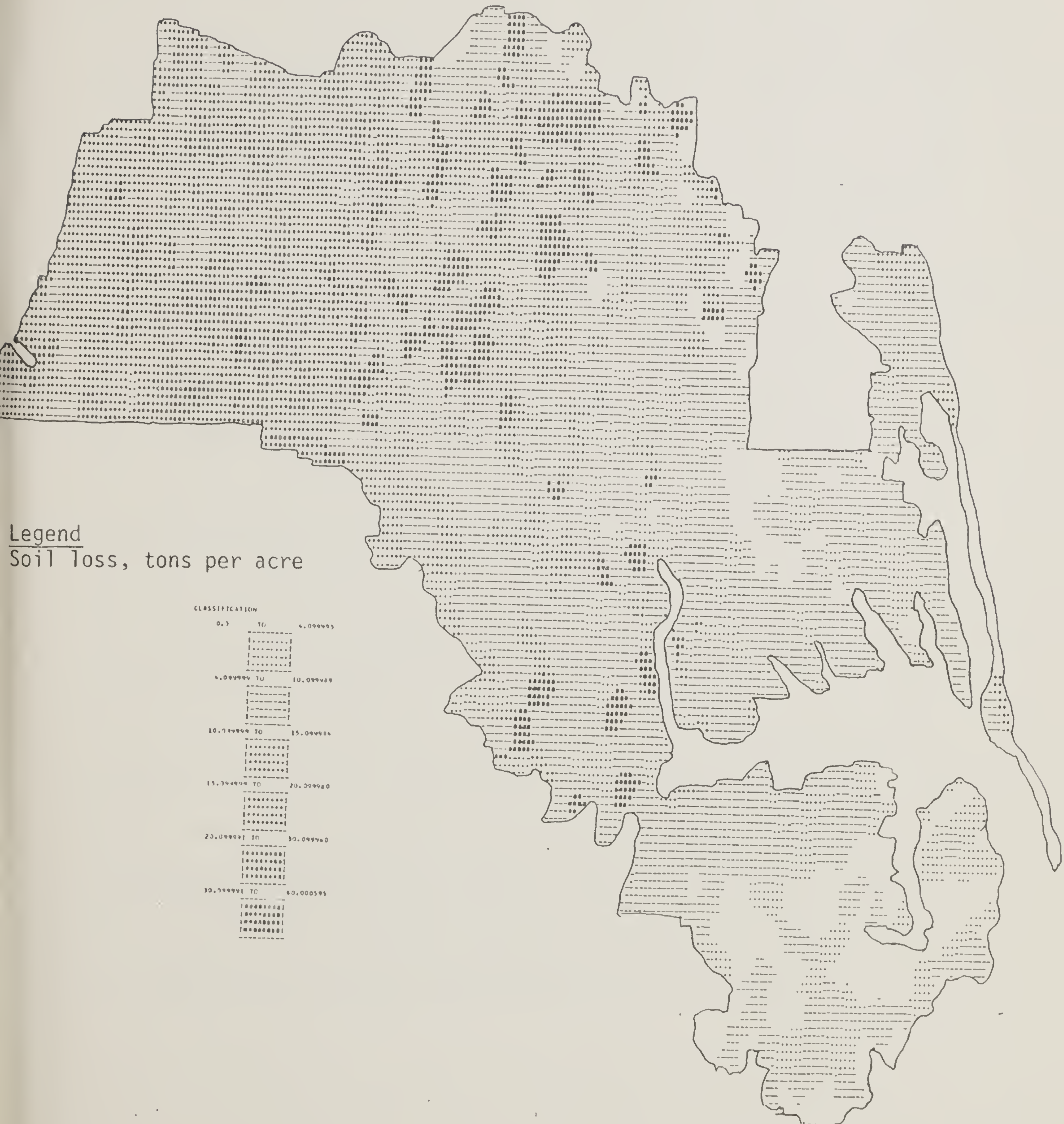
2/ Field stripcropping with meadow strips.

3/ Field stripcropping with small grain winter cover.

4/ Contour stripcropping with meadow strips.

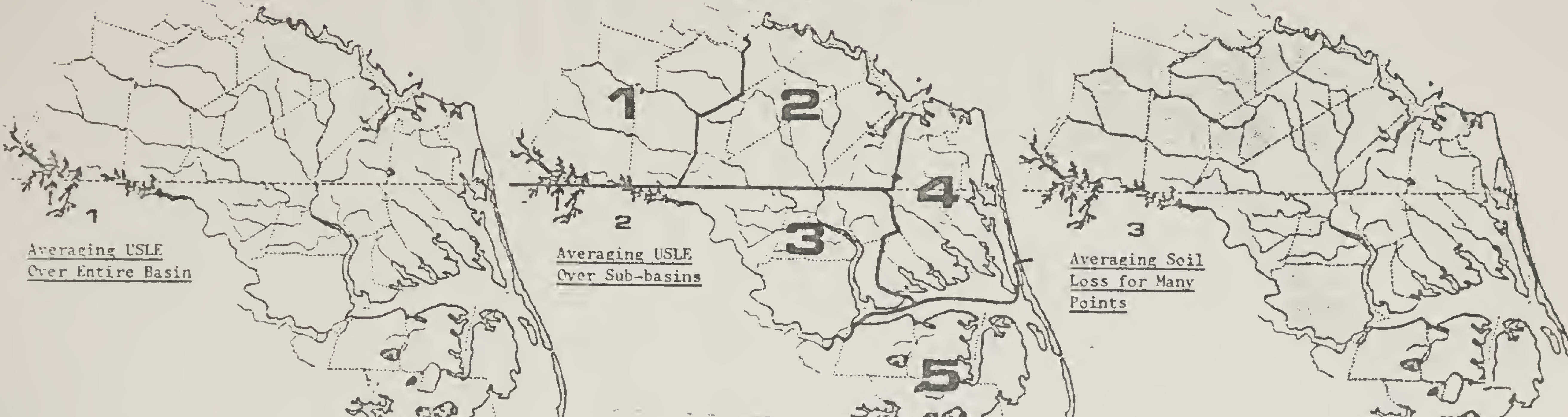
5/ Contour stripcropping with small grain winter cover.

Figure 9 --Location of cropland with various erosion rates projected for 1990



This method for obtaining gross soil erosion can be favorably compared with similar methods which also use the Universal Soil Loss Equation.) Factors are evaluated for each sample point and the soil losses averaged, rather than averaging the factor values. Figure 10 compares the method used in this study with two alternative methods for computing soil loss. The first method involves averaging the factor values of the USLE over the entire Basin, a process which naively overlooks huge differences in conditions over an area this large. Method 2 ameliorates these differences somewhat by averaging the USLE factors for sub-basins judged to be more homogeneous. The third method, used in this study, computes the soil loss for many points within the Basin and averages the soil loss for the soils making up each SPG rather than averaging the factor values. This method more completely accounts for differences in rainfall, soil, and slope over the Basin and is likely to be more accurate. For making current soil loss estimates or projections, this method should be even more acceptable when the results of the 1977 Soil Conservation Service Erosion Inventory are available. (SCS, June, 1977) This inventory resurveys a subset of the original sample points consisting of two to three sample points within specified PSU's. Land use and treatment information will be updated and information on specific types of crops grown, conservation practices used, and USLE factors, including slope length factors which were missing from the 1967 inventory, will be added.

Figure 10-- Three Methods for Computing Soil Loss, Chowan Basin



1 Averaging USLE Over Entire Basin

R = 286
K = .20
LS = 2.00
P = 1.0
C = .40

RESULTS:
Average Loss in Basin = 45.8 tons/ac
Total Loss in Basin = 64.2 million tons

2 Averaging USLE Over Sub-basins

	Sub-basin 1	2	3	4	5	
R =	225	257	302	328	350	
K =	.25	.20	.15	.15	.05	
LS =	2.46	1.94	1.86	1.29	0.67	
P =	1.0	1.0	1.0	1.0	1.0	
C =	.40	.40	.40	.40	.40	
Avg Loss =	55.4	39.9	33.7	25.4	4.7	tons/ac
Total Loss =	13.9	15.3	12.0	7.2	0.6	million tons

RESULTS:
Average Loss in Basin = 34.9 tons/ac
Total Loss in Basin = 48.9 million tons

3 Averaging Soil Loss for Many Points

Computed for points sampled in the 1967 CNI and averaged by SPG.

RESULTS:
Average Loss in Basin = 19.1 tons/ac
Total Loss in Basin = 26.8 million tons

Conclusion

The methods presented in this paper, in conjunction with the CNI sample data, offer a spatial dimension to agricultural resource analysis which has largely been absent in the past. The ability to determine and spatially display resource attributes opens up a wide range of possible applications to river basin planning and to other studies.

For river basin planning, these methods aid both the creation of plans and their evaluation. Identification of changing agricultural areas, in terms of new clearing or in terms of urban encroachment, can be made by comparing 1967 land use from the CNI to later land use data from a variety of sources. Analysis of soils on which change occurs can further focus planning efforts on those areas important to a basin's future. Impacts of alternative plans can be displayed graphically for easy comparison. The mapping programs presented here are designed to display any of 27 outputs from an LP model, located by means of SPG. These outputs include economic information such as cost of agricultural production, practice information such as amount of terracing, production information on each crop grown and environmental information such as amounts of fertilizer applied. The output of any other analysis by soil productivity groups can also be displayed. Because the method used stores the zone information created by the nearest-neighbor technique, each subsequent map is inexpensive to create. Changes in shading or interval values can be accommodated easily.

Possible applications exist for other studies as well. Water quality studies can benefit from a more precise computation of soil erosion, which is of importance in nitrogen and phosphorus movement as well as in terms of sediments. Analysis of farm production policies can benefit from spatial displays of their potential impact on the location of production. The cost and ease of application of these methods open up the possibility of conducting fairly detailed planning work at the state or regional level. This could benefit water quality, conservation and pesticide studies which have relatively short turn-around and staffing. Other analyses will likely be possible in the future, to the extent that update studies continue to build on the existing CNI data base, adding new and different data items. The potential for the CNI to become the basis for a national land and water resource data base certainly exists and is enhanced by development of these and other methods to utilize such a data base.

LITERATURE CITED

1. Anderson, James R., Ernest E. Hardy, John T. Roach, and Richard E. Witmer. "A Land Use and Land Cover Classification System for Use with Remote Sensor Data." USGS Professional Paper 964, U.S. G.P.O., Washington, D.C., 1976.

2. Center for Census Use Studies. "Special Program Information Tape Documentation, Program: SCANGEN." U.S. Bureau of the Census, Washington, D.C., May, 1976.

3. Laboratory for Computer Graphics and Spatial Analysis. "POLYVRT User's Manual." Version 1.1, Graduate School of Design, Harvard University, Cambridge, Mass., November, 1974.

4. McElroy, A.D., S.Y. Chiu, V.W. Nebgen, A. Aleti, and F.W. Bennet. "Loading Functions for Assessment of Water Pollution from Nonpoint Sources." EPA 600/2-76-151, Environmental Protection Agency, Office of Research and Development, Washington, D.C., May, 1976.

5. Parks, John R. "Soil Productivity Groups, North Carolina." USDA, ERS, Raleigh, North Carolina, April, 1975.

6. U.S. Department of Agriculture, Soil Conservation Service. "The Universal Soil Loss Equation with Factor Values for North Carolina." Technical Guide Section II-D, Raleigh, North Carolina, October, 1976.

7. ______________________________, Soil Conservation Service. "Erosion Inventory; Instructions for the PSU and Point Data Collection." Washington, D.C., June, 1977.

8. ______________________________, Soil Conservation Service. Conservation Needs Memorandum-21, Washington, D.C., November 3, 1965.

9. ______________________________, Soil Conservation Service. Soils Memorandum-21 (Revision 2), Washington, D.C., November 10, 1965.

10. ______________________________, Soil Conservation Service. "Procedure for Computing Sheet and Rill Erosion on Project Areas." Technical Release No. 51 (Revision 2), Washington, D.C., September, 1977.

11. Wischmeier, Walter H. and Dwight D. Smith. "Predicting Rainfall-Erosion Losses from Cropland East of the Rocky Mountains." Agricultural Handbook No. 282, USDA, ARS, U.S.G.P.O., Washington, D.C., May, 1965.

Appendix A-1

CHOWMAP1

A computer mapping program that produces county outline maps with PSU's identified by SPG on the CALCOMP plotter.

Inputs-

Unit 8 U.P006066.CHPWAN.POLYVRT.CHAINS--chain file of county boundary segment chains.

Unit 9 U.P006066.CHOWANNC.CNICOORD.PSUSORT
U.P006066.CHOWANVA.CNICOORD.PSUSORT
coordinates and identifying data for PSU's in North Carolina and Virginia portions of the Chowan Basin.

Outputs-

Description of area mapped, window of map and points plotted.

Plotter map (see Figure 5).

Run Time-

Two quadrangle maps

Rocky Mount, NC

172 vertices in outline
165 data points (PSUs)

Northern Albemarle Sound, NC

465 vertices in outline
830 data points (PSUs)

8.0 seconds of IBM 370/168 CPU

```
1     //CHOWMAP JOB (006066,0047,#3035),'158 TOM ELWERTOWSKI'              103 3333
      ***SETUP PLOT,2400,RING=YES,DEFER=YES
      ***MESSAGE  ************* CALCOMP PLOT JOB  ****************************
      ***MESSAGE  ID='158 TOM ELWERTOWSKI'     PROJ #=006066       BIN #=0047
      ***MESSAGE  JOBNAME=CHOWMAP               LENGTH=6FT.
      ***MESSAGE  ***************************************************************
2     //DRAW EXEC PLOTCG
3     XXPLOT     PROC TAPE=PLOT                                                00010000
      ***                                                                      00020000
      ***              THE UNI-COLL CALCOMP PLOTTER LIBRARY                    00030000
      ***                                                                      00040000
      ***        PROCEDURE TO COMPILE, LOAD AND EXECUTE CALCOMP PLOTTER        00050000
      ***        SOURCE MODULES.                                               00060000
      ***                                                                      00070000
4     XXFORT     EXEC PGM=FRTGI,REGION=130K                                    00080000
      ***                                                                      00090000
5     XXCONTROLC DD  DSN=SYS6.CONTROL(TGIFORT),DISP=SHR                        00100000
6     XXMSGPRINT DD  SYSOUT=A                                                  00110000
7     XXSYSLIN   DD  DSN=&LOADSET,UNIT=SYSSQ,DISP=(MOD,PASS,DELETE),           00120000
      XX             DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),                     00130000
      XX             SPACE=(CYL,(1,1),,CONTIG)                                 00140000
8     XXSYSPRINT DD  SYSOUT=A                                                  00150000
9     XXSYSPUNCH DD  SYSOUT=B                                                  00160000
      ***                                                                      00170000
      ***                                                                      00180000
10    //SYSIN      DD *                  GENERATED STATEMENT
11    XXGO       EXEC PGM=LOADER,PARM=(MAP,PRINT),REGION=130K,                 00190000
      XX             COND=(4,LT,FORT)                                          00200000
      ***                                                                      00210000
12    XXFT04F001 DD  SYSOUT=A,DCB=(RECFM=FA,BLKSIZE=133)                       00220000
13    XXFT05F001 DD  DDNAME=SYSIN                                              00230000
14    XXFT06F001 DD  SYSOUT=A                                                  00240000
15    XXFT07F001 DD  SYSOUT=B                                                  00250000
16    XXSYSLIB   DD  DSN=SYS6.USERLIB,DISP=SHR                                 00260000
17    XX         DD  DSN=SYS6.USERLIB,DISP=SHR                                 00270000
18    XX         DD  DSN=SYS6.FORTLIB,DISP=SHR                                 00280000
19    XX         DD  DSN=SYS7.PLOTLIB,DISP=SHR                                 00290000
20    XX         DD  DSN=SYS7.FORTLIB,DISP=SHR                                 00300000
21    XXSYSLIN   DD  DSN=*.FORT.SYSLIN,DISP=(OLD,DELETE)                       00310000
22    XX         DD  DDNAME=OBJ                                                00320000
23    XXSYSLOUT  DD  SYSOUT=A                                                  00330000
24    XXPLOTTAPE DD  DSN=&TAPE,UNIT=(2400,,DEFER),VOL=SER=PLOT,DISP=OLD        00340000
25    //GO.FT08F001 DD DSN=U.P006066.CHOWAN.POLYVRT.CHAINS,DISP=SHR
26    //GO.FT09F001 DD DSN=U.P006066.CHOWANNC.CNICOORD.FSUSORT,DISP=SHR
27    //             DD DSN=U.P006066.CHOWANVA.CNICOORD.FSUSORT,DISP=SHR
28    //SYSIN      DD *                  GENERATED STATEMENT
```

```
      REAL COORD(2,100)
      DATA VOX/0./,VOY/0./
      CALL WINDOW(-09.0,-14.75,29.5)
      CALL START(PLTBUF,3000)
      CALL PLOT(-3.0,-14.75,23)
   10 READ (5,510,END=990) TITLE
  510 FORMAT(10X,10A4)
      WRITE(6,515) TITLE
  515 FORMAT('0MAP TITLE: ',10A4)
C           GET WINDOW LIMITS IN PAGE COORDINATES
      READ(5,520) WXL,WYB,WXR,WYT
  520 FORMAT(10X,4F10.0)
      WRITE(6,525) WXL,WYB,WXR,WYT
  525 FORMAT(' WINDOW:',4F10.2)
C           GET VIEWPORT SIZE IN SCREEN COORDINATES,
C             ALSO PLACEMENT OF VIEWPORT ON SCREEN
      READ(5,520) VSX,VSY,VXL,VYB
      WRITE(6,535) VSX,VSY,VXL,VYB
  535 FORMAT(' VIEWPORT:',2F7.2,'  AT:',2F7.2)
C           RESET ORIGIN TO LL CORNER OF VIEWPORT
      VXL=VXL-VOX
      VYB=VYB-VOY
      CALL PLOT(VXL,VYB,-3)
      VOX=VOX+VXL
      VOY=VOY+VYB
      CALL RECT(0.0,0.0,ABS(VSY),ABS(VSX),0.0,3)
C           SCALING AND TRANSLATION FACTORS
C           REFLECTION ABOUT X-AXIS BUILT-IN
      A=VSX/(WXR-WXL)
      D=VSY/(WYT-WYB)
      E=-WXR
      F=-WYB
      CALL SETCLP(WXL,WYB,WXR,WYT)
      CALL SETSCA(A,E,D,F)
C           PROCESS HEADER RECORDS ON CHAIN FILE
      READ(8) NCHAIN,NPOLYG,NVERT,NNODE,NLEV,NID,KORGF,KCOPDT,NOCC
      WRITE(6,555) NCHAIN,NPOLYG,NVERT
  555 FORMAT('0INPUT FILE OPENED.'/
     >' CHAINS=',I4,'  POLYGONS=',I4,'  VERTICES=',I5)
      IF(NID.GT.1) READ(8)
      IF(NOCC.GT.0) READ(8)
C           PROCESS INDIVIDUAL CHAINS
      DO 110 I=1,NCHAIN
      READ(8) ID,NPNT,NFROM,NTO,ILEFT,IRGH
      IF (NPNT.LE.100) GO TO 102
      WRITE(6,560) ID,NPNT,NFROM,NTO,ILEFT,IRGH
  560 FORMAT ('0CHAIN ENCOUNTERED WITH MORE THAN A HUNDRED POINTS.  ONLY
     > THE FIRST 100 HAVE BEEN ACCEPTED.'/
     >'   CHAIN NAME   POINTS     FROM (NODE) TO    LEFT(POLYGON)RIGHT'/
     >1X,6I10)
      NPNT=100
  102 READ(8) (COORD(1,J),COORD(2,J),IDET(J),J=1,NPNT)
      CALL CLIP(COORD,NPNT)
  110 CONTINUE
      CALL DRWLST(NVERT)
      WRITE(6,565) NVERT
  565 FORMAT(' VERTICES PLOTTED=',I5)
```

```
          NPNT=0
C              GET AND PLOT INDIVIDUAL POINTS
      210 READ(9,610,END=220) FPN,LAT1,LAT2,LNG1,LNG2
      610 FORMAT(35X,F3.0,17X,4I3)
          X=LNG1+LNG2/600.
          Y=LAT1+LAT2/600.
          IF (X.LT.WXL.OR.X.GT.WXR.OR.Y.LT.WYB.OR.Y.GT.WYT) GO TO 210
          NPNT=NPNT+1
          X=(X+E)*A
          Y=(Y+F)*D
          CALL SYMBOL(X,Y,0.05,4,0.0,-1)
          CALL NUMBER(X+0.05,Y-0.05,0.1,FPN,0.0,-1)
          GO TO 210
      220 WRITE(6,575) NPNT
      575 FORMAT('0DATA POINT FILE ACCESSED.'/
         >' VERTICES PLOTTED:',I5)
          CALL SYMBOL(0.0,VSY+0.5,0.5,TITLE,0.0,40)
          REWIND 8
          REWIND 9
          GO TO 10
      990 CALL WHERE(X,Y,1)
          CALL PLOT(X,Y,999)
          STOP
          END
```

```
      INTEGER C1,C2
      LOGICAL*1 B11,B12,B13,B14,B21,B22,B23,B24,B(8)
      EQUIVALENCE (C1,B11,B(1)),(B12,B(2)),(B13,B(3)),(B14,B(4)),
     >            (C2,B21,B(5)),(B22,B(6)),(B23,B(7)),(B24,B(8))
C          COMPUTE POSITION OF FIRST POINT RELATIVE TO WINDOW
      X1=COORD(1,1)
      Y1=COORD(2,1)
      B14=X1.LT.XL
      B13=X1.GT.XR
      B12=Y1.LT.YB
      B11=Y1.GT.YT
C          FOR EACH ADDITIONAL POINT,
C            COMPUTE POSITION RELATIVE TO SCREEN
C            DETERMINE STATUS OF LINE CONNECTING THIS POINT AND THE LAST
      DO 90 I=2,NPNT
      X2=COORD(1,I)
      Y2=COORD(2,I)
      B24=X2.LT.XL
      B23=X2.GT.XR
      B22=Y2.LT.YB
      B21=Y2.GT.YT
C          IS LINE COMPLETELY VISIBLE?
   10 IF (C1.EQ.0.AND.C2.EQ.0) GO TO 70
C          IS LINE COMPLETELY INVISIBLE?
      IF (B11.AND.B21.OR.B12.AND.B22.OR.B13.AND.B23.OR.B14.AND.B24)
     >GO TO 80
      IF (C1.NE.0) GO TO 20
C          EXCHANGE POINTS 1 AND 2
      T=X1
      X1=X2
      X2=T
      T=Y1
      Y1=Y2
      Y2=T
      T=C1
      C1=C2
      C2=T
   20 IF (.NOT.B14) GO TO 30
C          PUSH TOWARD LEFT EDGE
      Y1=Y1+(Y2-Y1)*(XL-X1)/(X2-X1)
      X1=XL
      GO TO 60
   30 IF (.NOT.B13) GO TO 40
C          PUSH TOWARD RIGHT EDGE
      Y1=Y1+(Y2-Y1)*(XR-X1)/(X2-X1)
      X1=XR
      GO TO 60
   40 IF (.NOT.B12) GO TO 50
C          PUSH TOWARD BOTTOM EDGE
      X1=X1+(X2-X1)*(YB-Y1)/(Y2-Y1)
      Y1=YB
      GO TO 60
   50 CONTINUE
C          PUSH TOWARD TOP EDGE
      X1=X1+(X2-X1)*(YT-Y1)/(Y2-Y1)
      Y1=YT
```

```
   60 CONTINUE
C        RECOMPUTE C1
      B14=X1.LT.XL
      B13=X1.GT.XR
      B12=Y1.LT.YB
      B11=Y1.GT.YT
      GO TO 10
   70 CALL DRAW(X1,Y1,X2,Y2)
   80 X1=XLST
      Y1=YLST
      C1=CLST
   90 CONTINUE
      RETURN
      ENTRY SETCLP(XL,YB,XR,YT)
      RETURN
      END
```

```
      SUBROUTINE DRAW(X1,Y1,X2,Y2)
      REAL COORD(2,102),STPDEL(4)
      EQUIVALENCE (COORD(1,101),STPDEL(1))
      DATA STPDEL/0.,0.,1.,1./,RLGNEG/ZC1300001/
      IF (NPTS.LT.100) GO TO 10
      CALL LINE(COORD(1,1),COORD(2,1),NPTS,2,0,64)
      NVERT=NVERT+NPTS
      XLST=COORD(1,100)
      YLST=COORD(2,100)
      NPTS=1
      COORD(1,1)=XLST
      COORD(2,1)=YLST
C          CHECK WHETHER THIS LINE HAS AN ENDPOINT IN COMMON WITH THE
C            LAST LINE.  IF SO, PLACE ONLY THE NEW POINT IN THE CHAIN
C            BEING BUILT.
   10 IF (.NOT.(X1.EQ.XLST.AND.Y1.EQ.YLST)) GO TO 20
      NPTS=NPTS+1
      COORD(1,NPTS)=(X2+E)*A
      COORD(2,NPTS)=(Y2+F)*D
      XLST=X2
      YLST=Y2
      RETURN
   20 IF (.NOT.(X2.EQ.XLST.AND.Y2.EQ.YLST)) GO TO 30
      NPTS=NPTS+1
      COORD(1,NPTS)=(X1+E)*A
      COORD(2,NPTS)=(Y1+F)*D
      XLST=X1
      YLST=Y1
      RETURN
      ENTRY DRWLST(NV)
      NV=NVERT+NPTS
C          IF YOU ARE HERE, THE LINE IS NOT CONNECTED TO THE PREVIOUS
C            ONE.  PLOT CHAIN ACCUMULATED THUS FAR AND BEGIN NEW CHAIN.
   30 COORD(1,NPTS+1)=0.
      COORD(2,NPTS+1)=0.
      COORD(1,NPTS+2)=1.
      COORD(2,NPTS+2)=1.
      IF (NPTS.NE.0) CALL LINE(COORD(1,1),COORD(2,1),NPTS,2,0,64)
      NVERT=NVERT+NPTS
      NPTS=2
      COORD(1,1)=(X1+E)*A
      COORD(2,1)=(Y1+F)*D
      COORD(1,2)=(X2+E)*A
      COORD(2,2)=(Y2+F)*D
      XLST=X2
      YLST=Y2
      RETURN
      ENTRY SETSCA(A,E,D,F)
      NPTS=0
      NVERT=0
      XLST=RLGNEG
      YLST=RLGNEG
      RETURN
      END
```

```
MAP TITLE: ROCKY MOUNT QUAD
WINDOW:       76.00      35.00      78.00      36.00
VIEWPORT: -28.75  17.50  AT:    0.0     2.00

INPUT FILE OPENED.
CHAINS=  30  POLYGONS=  26  VERTICES=  313
VERTICES PLOTTED=  172

DATA POINT FILE ACCESSED.
VERTICES PLOTTED:  165

MAP TITLE: NORTHERN ALBEMARLE SOUND
WINDOW:       76.00      36.00      78.00      37.00
VIEWPORT: -28.75  17.50  AT:  ~~22.00~~    2.00
                              32.00
INPUT FILE OPENED.
CHAINS=  30  POLYGONS=  26  VERTICES=  31
VERTICES PLOTTED=  465

DATA POINT FILE ACCESSED.
VERTICES PLOTTED:  830
```

Appendix A-2

CHOWMAP2

A collection of computer programs that produces chloropleth maps of zones based around PSUs on the line printer. The programs are:
MATMAP--produces a numerical representation of the map matrix called a scandeck which can be stored for reuse.

Inputs-

Unit 5 Control cards specifying the kind of map to be produced, the geographic coordinates, the size of the viewpoint, the characteristics of the printer, a map title and the input file format.

Unit 8 Description of the input file including number of chains, number of polygons, number of vertices, and the number of nodes. The input file itself is also read on this unit and consists of chains of boundary segments.

Unit 9 Input file of data points which will form the zone centroids.

Outputs-

Unit FTOUT Output file for scandeck. An example of a scandeck is included in this appendix.

REFORMAT -- Reformats data to be mapped and selects PSU coordinates for use in SCANGEN.

Inputs-

Unit 1 Data to be mapped, by SPG
Unit 5 Coordinates of map frame
Unit 8 Coordinates of PSUs

Outputs-

Unit 9 Reformated file containing PSU coordinates, SPG associated with the PSU and data value to be mapped associated with that PSU through the SPG.

SCANGEN (Center for Census Use Studies, May, 1976)--Actually produces the map.

Inputs-

Control cards specifying input file and output map characteristics.

Outputs-

A description of the map produced including a statistical summary of the mapped intervals and the printer map (see Figure 9).

```
//DISK JOB (006066.MRRR,:03/5),CLASS=W
//FLPTCOMP EXEC FRTHC,PARM.FORT='DECK,OPT(2)'
//SYSPUNCH DD DSN=MATMAP.FRT.OBJ,UNIT=SYSDA,DISP=(NEW,KEEP),
//   DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),SPACE=(TRK,(10,5),RLSE)
      INTEGER TITLE(20),CMD,CMDLST(4),FTIN,FTOUT
      REAL PARAM(4)
      DATA NCMD/4/,CMDLST/'WIND','VIEW','LPIN','GENE'/
      WRITE(6,01)
   01 FORMAT(1H1,50X,'MATRIX MAPPING SYSTEM, V1M0')
  110 READ(5,02,END=990) CMD,PARAM
   02 FORMAT(A4,6X,4F10.0)
      DO 120 I=1,NCMD
      IF (CMD .EQ. CMDLST(I)) GO TO (10,20,30,40),I
  120 CONTINUE
      WRITE(6,05) CMD,PARAM
   05 FORMAT(' UNDECIPHERABLE COMMAND: ',A4,6X,4F10.4)
   10 WXL = PARAM(1)
      WYB = PARAM(2)
      WXR = PARAM(3)
      WYT = PARAM(4)
      WRITE(6,15) WXL,WYB,WXR,WYT
   15 FORMAT(' WINDOW: ',4F10.4)
      GO TO 110
   20 VSX = PARAM(1)
      VSY = PARAM(2)
      WRITE(6,25) VSX,VSY
   25 FORMAT(' VIEWPORT: ',2F10.4)
      GO TO 110
   30 LPINCH = PARAM(1) + 0.5
      WRITE(6,35) LPINCH
   35 FORMAT(' LINES PER INCH: ',I2)
      RPINCH = LPINCH
      CPINCH = 10.
      GO TO 110
   40 MAPTYP = PARAM(1) + 0.5
      FTOUT = PARAM(2) + 0.5
      FTIN = PARAM(3) +0.5
      WRITE(6,45) MAPTYP,FTOUT,FTIN
   45 FORMAT('0GENERATE MAP TYPE ',I2,', OUTPUT FILE ',I2,', BASE MAP FI
     1LE ',I2)
      READ(5,46) TITLE
   46 FORMAT(20A4)
      WRITE(6,47) TITLE
   47 FORMAT(' TITLE: ',20A4)
      DO 130 I = 1,3
      IF (MAPTYP .EQ. I) GO TO (130,420,430),I
  130 CONTINUE
      WRITE(6,49) MAPTYP
   49 FORMAT(' INVALID MAP TYPE ',I2)
      GO TO 110
  420 CALL SCALE(WXL,WYB,WXR,WYT,VSX,VSY,CPINCH,RPINCH,A,E,D,F)
      CALL SETCLP(WXL,WYB,WXR,WYT)
      CALL SETSCA(A,E,D,F)
      CALL RCHPRX
      NROW = LPINCH * ABS(VSY) + 0.5
      NCOL = CPINCH * ABS(VSX) + 0.5
      NZON = 2
      CALL SGEN(NROW,NCOL,NZON,FTOUT)
      GO TO 110
  430 CALL SCALE(WXL,WYB,WXR,WYT,VSX,VSY,CPINCH,RPINCH,A,E,D,F)
      NROW = LPINCH * ABS(VSY) + 0.5
      NCOL = CPINCH * ABS(VSX) + 0.5
```

```
      NZON = 2
      CALL RCNTRD(FTIN,FTOUT,NROW,NCOL,NZON,WXL,WYB,WXR,WYT,A,E,D,F)
      GO TO 110
  990 STOP
      END
      SUBROUTINE SCALE(WXL,WYB,WXR,WYT,VSX,VSY,CPINCH,RPINCH,A,E,D,F)
      A = VSX / (WXR - WXL) * CPINCH
      E = -WXL
      IF (VSX .LT. 0) E = -WXR
      D = VSY / (WYT - WYB) * RPINCH
      F = -WYB
      IF (VSY .LT. 0) F = -WYT
      RETURN
      END
      SUBROUTINE RCHPRX
      INTEGER IDET(100)
      REAL COORD(2,100)
      READ(8) NCHAIN,NPOLYG,NVERT,NNODE,NLEV,NID,KORGF,KCORDT,NDOC
      WRITE(6,15) NCHAIN,NPOLYG,NVERT
   15 FORMAT('0INPUT FILE OPENED.'/
     1' CHAINS=',I4,'  POLYGONS=',I4,'  VERTICES=',I5)
      IF (NID .GT. 1) READ(8)
      IF (NDOC .GT. 0) READ(8)
      DO 130 I = 1,NCHAIN
      READ(8) ID,NPNT,NFROM,NTO,ILFT,IRCH
      IF (ILFT .EQ. 0 .OR. IRGH .EQ. 0) GO TO 110
      READ(8)
      GO TO 130
  110 IF (IRGH .NE. 0) IRGH = 1
      IF (ILFT .NE. 0) ILFT = 1
      IF (IRGH .EQ. 0) IRGH = 2
      IF (ILFT .EQ. 0) ILFT = 2
      IF (NPNT .LE. 100) GO TO 120
      WRITE(6,25) ID,NPNT,NFROM,NTO,ILFT,IRCH
   25 FORMAT('0CHAIN ENCOUNTERED WITH MORE THAN A HUNDRED POINTS.  ONLY
     1THE FORST 100 HAVE BEEN ACCEPTED.'/
     2'   CHAIN NAME   POINTS      FROM (NODE) TO   LEFT(POLYGON)RIGHT'/
     31X,6I10)
      NPNT = 100
  120 READ(8) (COORD(1,J),COORD(2,J),IDET(J),J=1,NPNT)
      CALL CLIP(COORD,NPNT,ILFT,IRGH)
  130 CONTINUE
  140 READ(8,END=990)
      GO TO 140
  990 RETURN
      END
      SUBROUTINE CLIP(COORD,NPNT,ILFT,IRGH)
      REAL COORD(2,NPNT)
      INTEGER C1,C2,CLST
      LOGICAL*1 B11,B12,B13,B14,B21,B22,B23,B24,B(8),EXCHPT
      EQUIVALENCE (C1,B11,B(1)),(B12,B(2)),(B13,B(3)),(B14,B(4)),
     1            (C2,B21,B(5)),(B22,B(6)),(B23,B(7)),(B24,B(8))
      GO TO 100
      ENTRY SETCLP(XL,YB,XR,YT)
      RETURN
  100 CONTINUE
C           COMPUTE POSITION OF FIRST POINT RELATIVE TO WINDOW
      X1 = COORD(1,1)
      Y1 = COORD(2,1)
      B14 = X1 .LT. XL
      B13 = X1 .GT. XR
      B12 = Y1 .LT. YB
      B11 = Y1 .GT. YT
```

```
C           FOR EACH ADDITIONAL POINT,
C             COMPUTE POSITION RELATIVE TO SCREEN
C             DETERMINE STATUS OF LINE CONNECTING THIS POINT TO THE LAST
      DO 90 I = 2,NPNT
      X2 = COORD(1,I)
      Y2 = COORD(2,I)
      B24 = X2 .LT. XL
      B23 = X2 .GT. XR
      B22 = Y2 .LT. YB
      B21 = Y2 .GT. YT
      EXCHPT = .FALSE.
C           IS LINE COMPLETELY VISIBLE?
   10 IF (C1 .EQ. 0 .AND. C2 .EQ. 0) GO TO 70
C           IS LINE COMPLETELY INVISIBLE?
      IF (B11 .AND. B21 .OR. B12 .AND. B22 .OR. B13 .AND. B23 .OR.
     1    B14 .AND. B24) GO TO 80
      IF (C1 .NE. 0) GO TO 20
C           EXCHANGE POINTS 1 AND 2
      T = X1
      X1 = X2
      X2 = T
      T = Y1
      Y1 = Y2
      Y2 = T
      T = C1
      C1 = C2
      C2 = T
      EXCHPT = .NOT. EXCHPT
   20 IF (.NOT. B14) GO TO 30
C           PUSH TOWARD LEFT EDGE
      Y1 = Y1 + (Y2 - Y1) * (XL - X1) / (X2 - X1)
      X1 = XL
      GO TO 60
   30 IF (.NOT. B13) GO TO 40
C           PUSH TOWARD RIGHT EDGE
      Y1 = Y1 + (Y2 - Y1) * (XR - X1) / (X2 - X1)
      X1 = XR
      GO TO 60
   40 IF (.NOT. B12) GO TO 50
C           PUSH TOWARD BOTTOM EDGE
      X1 = X1 + (X2 - X1) * (YB - Y1) / (Y2 - Y1)
      Y1 = YB
      GO TO 60
   50 CONTINUE
C           PUSH TOWARD TOP EDGE
      X1 = X1 + (X2 - X1) * (YT - Y1) / (Y2 - Y1)
      Y1 = YT
   60 CONTINUE
C           RECOMPUTE C1
      B14 = X1 .LT. XL
      B13 = X1 .GT. XR
      B12 = Y1 .LT. YB
      B11 = Y1 .GT. YT
      GO TO 10
C           REORDER POINTS, IF THEY HAVE BEEN SWITCHED
   70 IF (.NOT. EXCHPT) GO TO 75
      T = X1
      X1 = X2
      X2 = T
      T = Y1
      Y1 = Y2
      Y2 = T
      T = C1
```

```
      C1 = C2
      C2 = T
   75 CALL DRAW(X1,Y1,X2,Y2,ILFT,IRGH)
   80 X1 = X2
      Y1 = Y2
      C1 = C2
   90 CONTINUE
      RETURN
      END
      SUBROUTINE DRAW(X1,Y1,X2,Y2,ILFT,IRGH)
      INTEGER XTO,YTO,XFROM,YFROM,TEMP,XLST,YLST
     1        ,DELTAX,DELTAY,DELTO2
      INTEGER*2 XYLR
      COMMON /WORK/ NPTS,NPTMAX,XYLR(4,5000)
      GO TO 100
      ENTRY SETSCA(A,E,D,F)
      DATA RLGNEG/Z80FFFFFF/
      NPTS = 0
      NPTMAX = 5000
      XL = RLGNEG
      YL = RLGNEG
      RETURN
  100 CONTINUE
C     CHECK WHETHER THIS LINE HAS ITS FIRST POINT IN COMMON WITH THE
C     LAST LINE'S SECOND POINT. IF SO, THERE IS NO NEED TO TRANSFORM
C     THE FIRST POINT SINCE THE INFO ALREADY EXISTS.
      IF (X1 .EQ. XL .AND. Y1 .EQ. YL) GO TO 110
      XFROM = (X1 + E) * A + 1.5
      YFROM = (Y1 + F) * D + 1.5
      GO TO 120
  110 XFROM = XLST
      YFROM = YLST
  120 XL = X2
      YL = Y2
C     TRANSFORM THE SECOND POINT.
      XTO = (X2 + E) * A + 1.5
      YTO = (Y2 + F) * D + 1.5
      XLST = XTO
      YLST = YTO
      JRGH = IRGH
      JLFT = ILFT
      IF (YFROM .LT. YTO) GO TO 130
C     IGNORE LINES WITH ZERO SLOPE.
      IF (YFROM .EQ. YTO) GO TO 150
C     SWITCH (XFROM,YFROM) , (XTO,YTO) SO THAT YFROM < YTO
      TEMP = YFROM
      YFROM = YTO
      YTO = TEMP
      TEMP = XFROM
      XFROM = XTO
      XTO = TEMP
      JRGH = ILFT
      JLFT = IRGH
C     CALCULATE STEPPING PARAMETERS
  130 DELTAX = XTO - XFROM
      DELTAY = YTO - YFROM
      DELTO2 = DELTAY / 2
      ISTEP = DELTAX / DELTAY
      IRMDR = MOD(DELTAX,DELTAY)
      IACCUM = 0
      IX = XFROM
      IBUMP = ISIGN(1,DELTAX)
      IRMDR = IABS(IRMDR)
```

```
      IF (NPTS + DELTAY .LE. NPTMAX) GO TO 132
      WRITE(6,05) NPTMAX
   05 FORMAT('0STORAGE LIMIT OF ',I5,' POINTS EXCEEDED. PROGRAM TERMINAT
     1ED.')
      STOP
C     OUTPUT A RASTER BREAKPOINT FOR EACH LINE
  132 NPTS = NPTS + 1
      XYLR(1,NPTS) = YFROM
      XYLR(2,NPTS) = IX
      XYLR(3,NPTS) = JRGH
      XYLR(4,NPTS) = JLFT
      IFR = YFROM + 1
      IF (IFR .EQ. YTO) GO TO 150
      ITO = YTO - 1
      DO 140 IROW = IFR,ITO
      IX = IX + ISTEP
      IACCUM = IACCUM + IRMDR
      IF (IACCUM .LT. DELTO2) GO TO 135
      IACCUM = IACCUM - DELTAY
      IX = IX + IBUMP
  135 NPTS = NPTS + 1
      XYLR(1,NPTS) = IROW
      XYLR(2,NPTS) = IX
      XYLR(3,NPTS) = JRGH
      XYLR(4,NPTS) = JLFT
  140 CONTINUE
  150 CONTINUE
      RETURN
      END
      SUBROUTINE SGEN(NROW,NCOL,NZON,FTOUT)
      INTEGER*2 XYLR,SCNLIN(2,130)
      INTEGER XYLRF(2,5000),FRST,FTOUT
      COMMON /WORK/ NPTS,NPTMAX,XYLR(4,5000)
      EQUIVALENCE (XYLR(1,1),XYLRF(1,1))
C     SORT POINTS BY 1. ROW, 2. COLUMN, 3. LEFT ZONE, 4. RIGHT ZONE
      CALL QSORTC(XYLR,NPTS)
C     ELIMINATE BREAKPOINTS FOR ZONES LESS THAN ONE CELL WIDE.
      NNPTS = 0
      FRST = 0
      XYLRF(1,NPTS+1) = 0
  120 IF (FRST .GE. NPTS) GO TO 190
      FRST = FRST + 1
      IF (XYLRF(1,FRST) .NE. XYLRF(1,FRST+1)) GO TO 123
      IF (XYLRF(2,FRST) .EQ. XYLRF(2,FRST+1)) GO TO 120
      GO TO 125
  123 NNPTS = NNPTS + 1
      XYLRF(1,NNPTS) = XYLRF(1,FRST)
      XYLRF(2,NNPTS) = XYLRF(2,FRST)
      GO TO 120
  125 IFR = FRST + 1
      DO 130 I = IFR,NPTS
      LAST = I
      IF (XYLRF(1,I) .NE. XYLRF(1,I+1)) GO TO 140
  130 CONTINUE
  140 IFR = FRST
      DO 170 I = IFR,LAST
      DO 150 J = FRST,LAST
      IF (XYLR(3,I) .EQ. XYLR(4,J)) GO TO 160
  150 CONTINUE
      GO TO 170
  160 XYLR(4,J) = XYLR(4,I)
      XYLRF(2,I) = XYLRF(2,FRST)
      FRST = FRST + 1
```

```
170 CONTINUE
    IF (FRST .GT. LAST) GO TO 182
    DO 180 I = FRST,LAST
    NNPTS = NNPTS + 1
    XYLRF(1,NNPTS) = XYLRF(1,I)
    XYLRF(2,NNPTS) = XYLRF(2,I)
180 CONTINUE
182 FRST = LAST
    GO TO 120
190 NPTS = NNPTS
C   CREATE SCAN LINE FOR EACH ROW OF THE MAP.
    FRST = 1
    DO 250 IROW = 1,NROW
    NZONR = 0
    LSTZON = NZON
    IF (XYLR(1,FRST) .GT. IROW .OR. FRST .GT. NPTS) GO TO 235
C   CHECK FOR UNIQUENESS OF THIS POINT.
    IF (XYLRF(1,FRST) .EQ. XYLRF(1,FRST+1)) GO TO 310
C   INSERT FIRST BREAKPOINT FOR THIS ROW IN SCAN LINE.
    NZONR = 1
    LSTZON = XYLR(4,FRST)
    SCNLIN(1,NZONR) = XYLR(3,FRST)
    SCNLIN(2,NZONR) = XYLR(2,FRST)
220 FRST = FRST + 1
C   CHECK FOR SAME ROW.
    IF (XYLR(1,FRST) .NE. IROW) GO TO 230
    IF (XYLRF(1,FRST) .EQ. XYLRF(1,FRST+1)) GO TO 310
C   CHECK WHETHER THIS POINT'S LEFT BOUNDRY MATCHES LAST ZONE'S RIGHT.
    IF (XYLR(3,FRST) .NE. LSTZON) GO TO 350
225 NZONR = NZONR + 1
    LSTZON = XYLR(4,FRST)
    SCNLIN(1,NZONR) = XYLR(3,FRST)
    SCNLIN(2,NZONR) = XYLR(2,FRST)
    GO TO 220
C   CREATE BREAKPOINT FOR LAST COLUMN, IF NECESSARY.
230 IF (SCNLIN(2,NZONR) .GE. NCOL) GO TO 240
235 NZONR = NZONR + 1
    SCNLIN(1,NZONR) = LSTZON
    SCNLIN(2,NZONR) = NCOL
C   OUTPUT SCAN LINE.
240 IF (NZONR .GT. 12) GO TO 242
    WRITE(FTOUT,50) IROW,NZONR,((SCNLIN(J,I),J = 1,2),I = 1,NZONR)
 50 FORMAT(26I3)
    GO TO 250
242 WRITE(FTOUT,52) IROW,NZONR,((SCNLIN(J,I),J = 1,2),I = 1,NZONR)
 52 FORMAT(26I3/(6X,24I3))
    GO TO 250
C   ERROR SECTION.
C   MORE THAN ONE BOUNDRY PAIR EXISTS FOR THIS POINT.
310 ITO = FRST + 1
    DO 320 I = ITO,NPTS
    LAST = I
    IF (XYLRF(1,I) .NE. XYLR(1,I-1)) GO TO 330
320 CONTINUE
330 WRITE(6,10) XYLR(1,FRST),XYLR(2,FRST),(XYLR(3,I),XYLR(4,I),
   1 I = FRST,LAST)
 10 FORMAT(' (',I3,',',I3,') HAS MORE THAN ONE (LEFT,RIGHT) PAIR.',
   1' THEY ARE ',7(' (',I4,',',I4,')'))
C   SEARCH FOR VALID POINT BY MATCHING LEFT SIDE ZONE WITH OLD RIGHT
C   SIDE ZONE.
    IF (NZONR .EQ. 0) GO TO 360
    DO 340 I = FRST,LAST
C   RETURN TO MAIN SECTION IF MATCH IS MADE.
```

```
      IF (XYLR(3,I) .EQ. LSTZON) GO TO 225
  340 CONTINUE
C     ONLY ONE POINT EXISTS BUT ITS LEFT SIDE DOES NOT MATCH THE LAST
C     ZONE'S RIGHT SIDE.
  350 WRITE(6,30) XYLR(1,FRST),XYLR(2,FRST),XYLR(3,FRST),LSTZON
   30 FORMAT(' (',I3,',',I3,') LEFT ZONE ',I4,' DOES NOT MATCH PREVIOUS
     1RIGHT ZONE ',I4)
  360 LF = FRST
      LL = LF
C     FETCH NEW POINT TO HELP RESOLVE PROBLEM.
  410 FRST = LL + 1
      LAST = FRST
      IF (XYLR(1,FRST) .EQ. XYLR(1,FRST-1)) GO TO 420
C     NO ADDITIONAL POINTS FOR THIS LINE.
      WRITE(6,20) LSTZON,XYLR(1,LF),XYLR(2,LF)
   20 FORMAT(' ZONE ',I4,' WILL BE EXTENDED PAST (',I3,',',I3,
     1') SINCE NO SUITABLE LEFT SIDE ZONE WAS FOUND.')
      GO TO 235
  420 IF (XYLRF(1,FRST) .NE. XYLRF(1,FRST+1)) GO TO 450
C     MORE THAN ONE BOUNDRY-PAIR EXIST FOR THIS POINT.
      ITO = FRST + 1
      DO 430 I = ITO,NPTS
      IF (XYLRF(1,I) .NE. XYLRF(1,I+1)) GO TO 440
  430 LAST = I
  440 WRITE(6,10) XYLR(1,FRST),XYLR(2,FRST),(XYLR(3,I),XYLR(4,I),
     1 I = FRST,LAST)
C     SEARCH FOR VALID POINT BY MATCHING THIS POINT'S LEFT ZONE WITH
C     THE LAST POINT'S RIGHT ZONE.
  450 DO 460 I = FRST,LAST
      DO 460 J = LF,LL
      IF (XYLR(3,I) .EQ. XYLR(4,J)) GO TO 470
  460 CONTINUE
C     NO MATCH FOUND.
C     OLD POINT IS IGNORED. NEW POINT BECOMES OLD POINT.
      WRITE(6,20) LSTZON,XYLR(1,LF),XYLR(2,LF)
      LF = FRST
      LL = LAST
      GO TO 410
  470 WRITE(6,40) XYLR(3,J),XYLR(1,J),XYLR(2,J),LSTZON
C     MATCH FOUND.
   40 FORMAT(' ZONE ',I4,' WILL BE TO THE LEFT OF (',I3,',',I3,
     1') THOUGH ZONE ',I4,' WAS ALSO A CANDIDATE.')
      NZONR = NZONR + 1
      SCNLIN(1,NZONR) = XYLR(3,J)
      SCNLIN(2,NZONR) = XYLR(2,J)
      GO TO 225
  250 CONTINUE
      IROW = 999
      WRITE(FTOUT,50) IROW
      RETURN
      END
      SUBROUTINE RCNTRD(FTIN,FTOUT,NROW,NCOL,NZONT,XL,YB,XR,YT,A,F,D,F)
      INTEGER CFMT(20),FTIN,FTOUT,YFRST,YLAST,TYFRST,TYLAST
      DATA RLP/77FFFFFF/
      INTEGER*2 SCNLI(2,130),SCNLO(2,130)
      COMMON /WORK/ NPTS,NPTMAX,CCORD(2,5000)
      READ(5,10) CFMT
   10 FORMAT(20A4)
      WRITE(6,15) CFMT
   15 FORMAT('0FORMAT FOR CENTROID FILE: ',20A4)
      NPTS = 0
  110 READ(9,CFMT,END=115) X1,Y1
      IF (X1 .LT. XL .OR. X1 .GT. XR .OR.
```

```
     1   Y1 .LT. YB .OR. Y1 .GT. YT) GO TO 110
      NPTS = NPTS + 1
      COORD(1,NPTS) = (Y1 + F) * D + 1.0
      COORD(2,NPTS) = (X1 + F) * A + 1.0
      GO TO 110
  115 WRITE(6,25) NPTS
   25 FORMAT(' CENTROID FILE OPENED.'/
     1' POINTS SELECTED=',I5)
C     SORT POINTS BY 1. ROW  2. COLUMN
      CALL GSORTC(COORD,NPTS)
C     COMPUTE SEARCH RADIUS
      SR = SQRT(NROW * NCOL / (NPTS * 3.14159))
      SR2 = SR * SR
      REWIND FTIN
      COORD(1,NPTS+1) = 999 + SR
      NZONO = NPTS + 1
      YFRST = 1
      YLAST = 1
C     INPUT A SCANLINE FROM OUTLINE FILE
  120 READ(FTIN,30) IROW,NZONRI,((SCNLI(I,J),I=1,2),J=1,12)
   30 FORMAT(26I3)
      IF (IROW .EQ. 999) GO TO 170
      IF (NZONRI .GT. 12) READ(FTIN,31) ((SCNLI(I,J),I=1,2),J=13,NZONRI)
   31 FORMAT(6X,24I3)
      YSMIN = IROW - SR
      YSMAX = IROW + SR
  130 IF (YSMIN .LE. COORD(1,YFRST)) GO TO 140
      YFRST = YFRST + 1
      GO TO 130
  140 IF (YLAST .EQ. NPTS) GO TO 150
      IF (YSMAX .LE. COORD(1,YLAST+1)) GO TO 150
      YLAST = YLAST + 1
      GO TO 140
  150 CONTINUE
      IZONRO = 0
      IZONL = 0
      IX = 0
C     PROCESS EACH SEGMENT OF INPUT SCANLINE.
      DO 270 J = 1,NZONRI
C     IF SEGMENT IS PART OF BACKGROUND, DO NOT INTERPOLATE.
      IF (SCNLI(1,J) .EQ. NZONI) GO TO 250
      JFR = IX + 1
      JTO = SCNLI(2,J)
C     ASSIGN EACH COLUMN OF SEGMENT TO NEAREST CENTROID.
      DO 240 IX = JFR,JTO
      XSMIN = IX - SR
      XSMAX = IX + SR
      IZON = 0
      D2MIN = BIG
C     EXAMINE EACH PIXEL.
      DO 220 I = YFRST,YLAST
      IF (COORD(2,I) .LT. XSMIN .OR. COORD(2,I) .GT. XSMAX) GO TO 220
C     COMPUTE DISTANCE TO CENTROID.
      D2 = (IX - COORD(2,I)) ** 2 + (IROW - COORD(1,I)) ** 2
      IF (D2 .GE. D2MIN) GO TO 220
      D2MIN = D2
      IZON = I
  220 CONTINUE
      IF (IZON .GT. 0 .AND. D2MIN .LE. SR2) GO TO 230
C     NO POINT FOUND WITHIN SEARCH RADIUS.  ENLARGE IT AND TRY AGAIN.
      TSR = SR
  310 TSR = TSR * 1.25
      TSR2 = TSR * TSR
```

```
      TYSMIN = IROW - TSR
      TYSMAX = IROW + TSR
      TYLAST = YLAST
  320 IF (TYFRST .EQ. 1) GO TO 330
      IF (TYSMIN .GT. COORD(1,TYFRST)) GO TO 330
      TYFRST = TYFRST - 1
      GO TO 320
  330 IF (TYLAST .EQ. NPTS) GO TO 340
      IF (TYSMAX .LE. COORD(1,TYLAST)) GO TO 340
      TYLAST = TYLAST + 1
      GO TO 330
  340 CONTINUE
      TXSMIN = IX - TSR
      TXSMAX = IX + TSR
      TYFRST = YFRST
      DO 350 I = TYFRST,TYLAST
      IF (COORD(2,I) .LT. TXSMIN .OR. COORD(2,I) .GT. TXSMAX) GO TO 350
      D2 = (IX - COORD(2,I)) ** 2 + (IROW - COORD(1,I)) ** 2
      IF (D2 .GE. D2MIN) GO TO 350
      D2MIN = D2
      IZON = I
  350 CONTINUE
      IF (IZON .GT. 0 .AND. D2MIN .LE. TSR2) GO TO 230
      GO TO 310
C     PIXEL HAS BEEN ASSIGNED TO A CENTROID.
  230 IF (IZON .EQ. IZONL) GO TO 240
      IF (IZONL .EQ. 0) GO TO 238
C     NEW ZONE. CREATE SEGMENT FOR OLD ONE.
      NZONRO = NZONRO + 1
      SCNLO(1,NZONRO) = IZONL
      SCNLO(2,NZONRO) = IX - 1
  238 IZONL = IZON
  240 CONTINUE
      IX = SCNLT(2,J)
      GO TO 270
C     BACKGROUND SEGMENT.  NO PIXELS ASSIGNED TO CENTROIDS.
C     GENERATE LAST INTERPOLATED SEGMENT, IF ANY.
  250 IF (IZONL .EQ. 0) GO TO 260
      NZONRO = NZONRO + 1
      SCNLO(1,NZONRO) = IZONL
      SCNLO(2,NZONRO) = IX
  260 IZONL = NZONO
      IX = SCNLT(2,J)
  270 CONTINUE
C     CREATE FINAL SEGMENT AND OUTPUT WHOLE SCANLINE.
      NZONRO = NZONRO + 1
      SCNLO(1,NZONRO) = IZONL
      SCNLO(2,NZONRO) = IX
  160 IF (NZONRO .GT. 12) GO TO 162
      WRITE(FTOUT,30) IROW,NZONRO,((SCNLO(I,J),I=1,2),J=1,NZONRO)
      GO TO 120
  162 WRITE(FTOUT,32) IROW,NZONRO,((SCNLO(I,J),I=1,2),J=1,NZONRO)
   32 FORMAT(26I3/(6X,24I3))
      GO TO 120
  170 WRITE(FTOUT,30) IROW
      RETURN
      END
//PLICOMP EXEC PLIOC,PARM.PLI='DECK,OPT(2)',
//SYSPUNCH DD DSN=MATMAP.PLI.OBJ,UNIT=SYSDA,DISP=(NEW,KEEP),
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200),SPACE=(TRK,(10,5),RLSE)
 QSORTC:PROC(K,N) OPTIONS(FORTRAN,NOMAP);
 /*  ALGORITHM QUIKSORT                                                  */
 /*  ADAPTED FROM TREMBLAY,J.P. AND P.G.SORENSON, " AN INTRODUCTION     */
```

```
/* TO DATA STRUCTURES WITH APPLICATIONS" (1976) PP.470-71 */
DCL K(5000) CHAR(A) CONNECTED,    N FIXED BIN(31),   T CHAR(A),
    (BOUND(2,15), LB, UB, TOP) FIXED BIN(15);
/*  1. INITIALIZE  */
TOP = 1;
BOUND(1,TOP) = 1;
BOUND(2,TOP) = N;
/*  2. PERFORM SORT  */
DO WHILE (TOP ¬= 0);
    /*  PROCESS NEW SUBFILE  */
    LB = BOUND(1,TOP);
    UB = BOUND(2,TOP);
    TOP = TOP - 1;
    DO WHILE (UB > LB);
        /*  3. INITIALIZATION PASS  */
        I = LB;
        J = UB;
        T = K(I);
       FOUR:/*  SCAN KEYS FROM RIGHT TO LEFT  */
        DO WHILE (T < K(J) & J > LB);
            J = J - 1;
            END;
        IF J <= I THEN DO;
            K(I) = T;
            GO TO SIX;
            END;
        ELSE DO;
            K(I) = K(J);
            I = I + 1;
            END;
        /*  5. SCAN KEYS FROM LEFT TO RIGHT  */
        DO WHILE (K(I) < T & I < UB);
            I = I + 1;
            END;
        IF J > I THEN DO;
            K(J) = K(I);
            J = J - 1;
            GO TO FOUR;
            END;
        ELSE DO;
            K(J) = T;
            I = J;
       SIX:/* PUSH DESCRIPTION OF UNPROCESSED SUBTABLE ON STACK  */
            END;
        TOP = TOP + 1;
        IF I-LB < UB-I THEN DO;
            BOUND(1,TOP) = I + 1;
            BOUND(2,TOP) = UB;
            UB = I - 1;
            END;
        ELSE DO;
            BOUND(1,TOP) = LB;
            BOUND(2,TOP) = I - 1;
            LB = I + 1;
            END;
        END;
    END;
END WSORTC;
//
```

Example of scandeck output--each line represents one row of the map matrix, indicated by the first number. The number of segments in the line (corresponding to the number of zones, including background areas, moving from left to right across the map matrix) is indicated by the second number. Pairs of numbers following this give the zone identification (internal to the program) and the ending print position for the zone.

Lines 1 through 13 below are all background (water) lines consisting of 1 zone identified as 958 which ends in print position 132. Line 14 is the first line in which some mapped area of interest shows up. It has 2 zones, the first identified as zone 4 which ends in print position 4, and the second which is background.

```
 1   1958132
 2   1958132
 3   1958132
 4   1958132
 5   1958132
 6   1958132
 7   1958132
 8   1958132
 9   1958132
10   1958132
11   1958132
12   1958132
13   1958132
14   2   4   4958132
15   5   4   5   1   9   2 11   3 12958132
16   7   4   6       9   2 11   3 13958 19   5 21958132
17   7   7   1   4   5   6   6   1   9   2 11   3 15958 17   5 22958132
18 10   7   2   4   5   6   7   1   9   2 11   3 14   6 17   9 18   5 23958132
19 10   7   3   4   5   6   6   1   9   2 11   3 14   6 17   9 19   5 23958132
20   9   7   4   4   5   6   9 11 13 12 14   6 17   9 20 13 22958132
21 11 10   1   7   4 18   5 14   9 11 13 12 16   6 17   9 19 13 22 17 23958132
```

```
      INTEGER SPG                                                       00000080
      REAL WOP(33),EROS(33)                                             00000090
  210 READ(1,30,END=220) SPG,WOP(SPG),EROS(SPG)                         00000100
   30 FORMAT(I2,2F4.1)                                                  00000110
      GO TO 210                                                         00000120
  220 CONTINUE                                                          00000130
      READ(5,10) XL,YB,XR,YT                                            00000140
 10   FORMAT(10X,4F10.0)                                                00000141
      WRITE(6,15) XL,YB,XR,YT                                           00000160
   15 FORMAT(' WINDOW:',4F10.4)                                         00000170
      NPTS = 0                                                          00000180
  110 READ(8,20,END=120) X1,Y1,SPG                                      00000190
   20 FORMAT(2F9.5,I3)                                                  00000200
      IF (X1 .LT. XL .OR. X1 .GT. XR .OR.                               00000210
     1    Y1 .LT. YB .OR. Y1 .GT. YT) GO TO 110                         00000220
      WRITE(9,22) X1,Y1,SPG,WOP(SPG),EROS(SPG)                          00000230
   22 FORMAT(2F9.5,I3,2F6.1)                                            00000240
      NPTS = NPTS + 1                                                   00000250
      GO TO 110                                                         00000260
  120 WRITE(6,25) NPTS                                                  00000270
   25 FORMAT(' POINTS SELECTED=',I5)                                    00000280
      STOP                                                              00000290
      END                                                               00000300

      /GO                                                               00000310
WINDOW:   77.0250   35.2500   78.8000   37.5000
POINTS SELECTED=  610

STATEMENTS EXECUTED=     4701

CORE USAGE      OBJECT CODE=     1072 BYTES,ARRAY AREA=      264 BYTES,TOTAL AREA AVAILABLE=  114688  BYTES

DIAGNOSTICS         NUMBER OF ERRORS=         0, NUMBER OF WARNINGS=       0, NUMBER OF EXTENSIONS=       0

COMPILE TIME=     0.03 SEC,EXECUTION TIME=     1.21 SEC,     12.01.25    THURSDAY     13 JUL 78    WATFIV - JUN 1977 V1L6

      C/STOP
```

```
* SCA[illegible] *  U.S. BUREAU OF CENSUS - C[illegible] STUDIES
* SCA[illegible] *     SCAN LINE GENERATOR FOR [illegible]
REFER[illegible]

USER [illegible] A LIST OF KEYWORD COMMANDS BY [illegible] IN COLUMN 1 WHILE IN KEYWORD MODE

KEYWORD PHRASES?
SCA[illegible] DIRECTLY FROM A SCAN [illegible] DATA ARRAY[illegible]

KEYWORD PHRASES?
RANGES([illegible],4.1,10.1,15.1,20.1,30.1,[illegible])                     00000397

KEYWORD PHRASES?
SHADE[illegible] YOUR OWN SHADING SYMBOLS                     00000410
ENTER [illegible] CHARACTERS IN COLUMNS 1 THRU [illegible]

MAP SHADING CHARACTERS?
.                                                             [illegible]

MAP SHADING CHARACTERS?
-                                                             [illegible]

MAP SHADING CHARACTERS?
+                                                             [illegible]

MAP SHADING CHARACTERS?
*                                                             [illegible]

MAP SHADING CHARACTERS?
X                                                             00000432

MAP SHADING CHARACTERS?
[illegible]                                                   00000461

MAP SHADING CHARACTERS?
END                                                           [illegible]00480

KEYWORD PHRASES?
MAPTITLE[illegible] UP [illegible] A SIX CARD TITLE           [illegible]

MAP TITLE CARD?

[illegible] RIVER BASIN                                       00000500

MAP TITLE CARD?

GROSS SOIL EROSION PER CROPPED ACRE                           00000501

MAP TITLE CARD?

WITHOUT PLAN SOLUTION                                         00000502

MAP TITLE CARD?

END                                                           00000530

KEYWORD PHRASES?
FORMAT[illegible] THE FOLLOWING CARD IS THE [illegible] DATA  00000540

FORMAT STATEMENT?

[illegible]                                                   [illegible]
```

```
    6  61099999     000                                                   0
CHOWAN-PASQUOTANK RIVER BASIN
GROSS SOIL EROSION PER CROPPED ACRE
WITHOUT PLAN SOLUTION

(21X,F6.1)                                                            00000550
        0.         4.        10.        15.        20.        30.        60.
.
-
+
*
><
OXVA
```

Appendix A-3

EROSION

A computer program which calculatres gross bare-ground soil loss using the universal soil loss equation with factors evaluated for each sample point in the Conservation Needs Inventory.

Inputs-

Unit 10 *.SORT.SORT.SORTOUT--individual CNI sample records sorted by soil productivity group.

Unit 11 U.P006066.RH.EROSION.DATA--Factor values of the USLE.

R factor by county

P factor by slope interval and practice

K and L factors by dominant soil

Output-

Table of normal and terraced gross bare-ground soil loss estimates by SPG and conservation practice, acreage over which computed and acreage-weighted average slope.

```
1    //EROSION JOB (6066$#,0047,#40$5),CLASS=A                               JOB  518
     ***ROUTE PRINT TSO
     ***ROUTE PUNCH TSO
     ***SETUP X21461,3400-4,IL=SCS623
2    //ONE EXEC PGM=QUIKJOB
3    //STEPLIB DD DSN=SYS6.QUIKJOB.VC,DISP=SHR
4    //SYSPRINT DD SYSOUT=A
5    //SYSUT1 DD DSN=CNI.NORTHCAR,VOL=SER=SCS623,LABEL=(2,SL),UNIT=3400-4,
     //         DCB=(RECFM=FB,DEN=2,LRECL=112,BLKSIZE=3360),DISP=(OLD,KEEP)
6    //SYSUT2 DD UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(100,20),RLSE),
     //          DCB=(RECFM=FB,LRECL=112,BLKSIZE=3360)
7    //SYSIN DD *
8    //SORT EXEC SORTUP
9    XXSORT     PROC PRIMARY=10                                               00010000
     ***                                                                      00020000
     ***              SORT/MERGE  VERSION 1 MODIFICATION LEVEL 4              00030000
     ***                                                                      00040000
     ***        PROCEDURE TO EXECUTE THE SORT/MERGE UTILITY PROGRAM FOR       00050000
     ***        STANDARD SORTS.                                               00060000
     ***                                                                      00070000
10   XXSORT     EXEC PGM=IERRC000,PARM='MSG=AP,LIST',REGION=100K              00080000
     ***                                                                      00090000
11   XXSORTLIB  DD  DSN=SYS6.SORTLIB,DISP=SHR                                 00100000
12   XXSORTWK01 DD  UNIT=SYSSQ,SPACE=(CYL,(&PRIMARY.),,CONTIG)                00110000
13   XXSORTWK02 DD  UNIT=SYSSQ,SPACE=(CYL,(&PRIMARY.),,CONTIG)                00120000
14   XXSORTWK03 DD  UNIT=SYSSQ,SPACE=(CYL,(&PRIMARY.),,CONTIG)                00130000
15   XXSORTWK04 DD  UNIT=SYSSQ,SPACE=(CYL,(&PRIMARY.),,CONTIG)                00140000
16   XXSORTWK05 DD  UNIT=SYSSQ,SPACE=(CYL,(&PRIMARY.),,CONTIG)                00150000
17   XXSORTWK06 DD  UNIT=SYSSQ,SPACE=(CYL,(&PRIMARY.),,CONTIG)                00160000
18   XXSYSOUT   DD  SYSOUT=A                                                  00170000
19   //SORTIN DD DSN=*.ONE.SYSUT2,UNIT=SYSDA,DISP=(OLD,PASS)
20   //SORTOUT DD  UNIT=SYSDA,DISP=(NEW,PASS),SPACE=(TRK,(100,20),RLSE),
     //        DCB=(RECFM=FB,LRECL=112,BLKSIZE=3360)
21   //SYSIN     DD *                  GENERATED STATEMENT
22   //THREE EXEC WATFIV
23   XXWATFIV  PROC  V=1,M=6,VI=6,MI=0,TYPE=SINGLE                           00010001
     ***                                                                      00020000
     ***              WATFIV  VERSION 1 MODIFICATION LEVEL 6        11/25/77 00030000
     ***                                                                      00040000
     ***    PROCEDURE TO EXECUTE THE WATERLOO FORTRAN LANGUAGE MONITOR        00050004
     ***                                                                      00060000
24   XXGO        EXEC PGM=WATFIV,REGION=256K                                  00070001
     ***                                                                      00080000
25   XXFT05F001 DD  DDNAME=SYSIN                                              00090000
26   XXFT06F001 DD  SYSOUT=A                                                  00100000
27   XXFT07F001 DD  SYSOUT=B                                                  00110000
28   XXSTEPLIB  DD  DSN=SYS6.WATFIV.V&V.L&M..JOBLIB,DISP=SHR                  00120000
29   XXWATLIB   DD  DSN=SYS6.WATFIV.USERLIB,DISP=SHR                          00130000
30   XX         DD  DSN=SYS6.WATFIV.V&V.L&M..LIBRARY,DISP=SHR                 00140003
31   XX         DD  DSN=SYS7.APPLMATH.IMSL.V&VI.M&MI..WATFIV.&TYPE.,          00150000
     XX             DISP=SHR                                                  00160000
32   //GO.FT10F001 DD DSN=*.SORT.SORT.SORTOUT,UNIT=SYSDA,DISP=(OLD,PASS)
33   //GO.FT11F001 DD DSN=U.P006066.RH.EROSION.DATA,UNIT=OLS,
     //             DISP=SHR
34   //SYSIN DD *
```

```
      PROGRAM FRES,TIME=[illegible],PAGES=[illegible]
      INTEGER [illegible] K(4,4),R(15,2)
      REAL [illegible],R(4),T[illegible](1)(1),P(4,4)
      WRITE(6,[illegible])
 [illegible] FORMAT('1STATE: NORTH CAROLINA')
      WRITE(6,[illegible])
 [illegible] FORMAT(' [illegible]')
      WRITE(6,[illegible])
 [illegible] FORMAT(' ',17X,'N[illegible]    [illegible]    [illegible]    [illegible]    [illegible]
     1 [illegible]    AREA    [illegible]')
      WRITE(6,[illegible])
 [illegible] FORMAT(' ',17X,'PRACTICE  FARMING  [illegible]    [illegible]    [illegible]
     1 [illegible]    COVERED  SLOPE')
 [illegible])
 [illegible]
 [illegible] [illegible](P(I,J),J=1,[illegible])
 [illegible] FORMAT([illegible].2)
 [illegible]
      [illegible](I,I[illegible])(P(I,J),J=1,2)
 [illegible] FORMAT([illegible],[illegible]I4)
 [illegible] [illegible].0
      [illegible].0
      [illegible].1
      [illegible].0
 [illegible]
 [illegible]
      [illegible]
 [illegible] [illegible].0
      READ(5,101)((K(I,J),[illegible]),J=1,4
 101  FORMAT(5X,15I4)
 [illegible] [illegible](I3),I3,[illegible],[illegible]),ITY,[illegible],[illegible]
 102  FORMAT(2X,I3,20X,I3,5X,I3,I3,I3,6X,I3,[illegible],F[illegible].1)
      IF(EP[illegible].[illegible].ISP,[illegible]) [illegible]
 [illegible] I=1,4
      IF([illegible].EQ.K(I,1)) GO TO 22
 [illegible]
 [illegible] 21
 22   KVAL=K(2,I)
      KVAL=K(3,I)
 [illegible] I=1,15
      IF(ITY.EQ.R(I,1)) GO TO [illegible]
 [illegible]
 [illegible] 21
 20   [illegible]=R(I,2)
      A=LVAL**0.5
      B=(.00763I+(.0053241*ISLOPE)+(.000763I*(ISLOPE**2)))
      ILS=A*B
      IS=IS+(ISLOPE*ACRES)
      [illegible]LS=[illegible]LS+([illegible]*ILS*KVAL*[illegible]*.01)
      [illegible]=[illegible]+ACRES
      IF(ISLOPE.GT.06.OR.ISLOPE.LT.02) GO TO [illegible]
      IF(ISLOPE.LT.[illegible]) GO TO 2[illegible]
      A=72**0.5
      GO TO 2[illegible]
 26   A=[illegible]**0.5
 27   ILS=A*B
      TAREA=TAREA+ACRES
      T[illegible]LS=T[illegible]LS+([illegible]*ILS*KVAL*ACRES*.01)
      GO TO 21
 23   [illegible]S=I[illegible]/AREA
      IF(IS.LE.[illegible]2) M=1
      IF(IS.GT.02.AND.IS.LE.07)M=2
      [illegible]
```

```
      IF(IS.GT.12.AND.IS.LE.18)M=4
      IF(IS.GT.18) M=5
      DO 6 J=1,6
       TRKLSP(J)=(P(J,M)*TRKLS)/TAREA
 6     FKLSP(J)=(P(J,M)*RKLS)/AREA
      CALL PRINT(RKLSP,TRKLSP,AREA,TAREA,IS,IPSPG)
      IPSPG=IPSPG+1
      BACKSPACE 10
      GO TO 24
998   WRITE(6,103)
103   FORMAT('OERROR IN ONE RECORD READ ON TAPE INPUT FILE 10')
       GO TO 21
999   STOP
      END

      SUBROUTINE PRINT(R,T,AR,TAR,ISL,ISP)
      DIMENSION R(6),T(6)
      WRITE(6,100)ISP,(R(J),J=1,6),AR,ISL
100   FORMAT('OSPG',I3,' NORMAL',4X,7F10.1,I5)
      WRITE(6,101)ISP,(T(J),J=1,6),TAR
101   FORMAT('OSPG',I3,' TERRACE',3X,7F10.1)
       RETURN
       END

 /GO
```

	NO PRACTICE	CONTOUR FARMING	FIELD SC A	FIELD SC B	CONTOUR, SC A	,	CONTOUR SC B	AREA COVERED	AVG SLOPE
SPG 10 NORMAL	39.9	23.9	14.0	18.0		12.0	16.0	4865.9	2
SPG 10 TERRACE	22.6	13.5	7.9	10.2		6.8	9.0	4865.9	
SPG 11 NORMAL	118.4	59.2	41.4	47.4		29.6	41.4	4865.9	7
SPG 11 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 12 NORMAL	0.0	0.0	0.0	0.0		0.0	0.0	0.1	0
SPG 12 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 13 NORMAL	20.4	12.3	7.2	9.2		6.1	8.2	6208.3	0
SPG 13 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 14 NORMAL	0.0	0.0	0.0	0.0		0.0	0.0	0.1	0
SPG 14 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 15 NORMAL	28.0	16.8	9.8	12.6		8.4	11.2	1390.3	0
SPG 15 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 16 NORMAL	0.0	0.0	0.0	0.0		0.0	0.0	0.1	0
SPG 16 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 17 NORMAL	0.0	0.0	0.0	0.0		0.0	0.0	0.1	0
SPG 17 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 18 NORMAL	42.2	25.3	14.8	19.0		12.7	16.9	152092.5	2
SPG 18 TERRACE	20.8	12.5	7.3	9.3		6.2	8.3	83767.6	
SPG 19 NORMAL	131.7	79.0	52.7	59.3		39.5	52.7	14574.0	8
SPG 19 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 20 NORMAL	135.7	67.8	47.5	54.3		33.9	47.5	60570.1	5
SPG 20 TERRACE	34.3	17.2	12.0	13.7		8.6	12.0	27622.2	
SPG 21 NORMAL	0.0	0.0	0.0	0.0		0.0	0.0	0.1	0
SPG 21 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 22 NORMAL	23.9	14.3	8.4	10.8		7.2	9.6	117354.5	2
SPG 22 TERRACE	12.7	7.6	4.4	5.7		3.8	5.1	62629.9	
SPG 23 NORMAL	45.9	27.5	16.1	20.6		13.8	18.3	209445.7	1
SPG 23 TERRACE	32.0	19.2	11.2	14.4		9.6	12.8	8088.9	
SPG 24 NORMAL	23.9	14.3	8.4	10.8		7.2	9.6	64201.3	0
SPG 24 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 25 NORMAL	26.8	16.1	9.4	12.1		8.0	10.7	76532.1	1
SPG 25 TERRACE	12.1	7.2	4.2	5.4		3.6	4.8	6406.5	
SPG 26 NORMAL	33.5	20.1	11.7	15.1		10.1	13.4	494347.6	0
SPG 26 TERRACE	32.5	19.5	11.4	14.6		9.7	13.0	107.3	
SPG 27 NORMAL	25.0	15.0	8.7	11.2		7.5	10.0	300781.5	0
SPG 27 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 28 NORMAL	19.8	11.9	6.9	8.9		5.9	7.9	23709.0	0
SPG 28 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 29 NORMAL	395.3	316.2	197.7	257.0		158.1	197.7	8153.0	16
SPG 29 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 30 NORMAL	57.9	29.0	20.3	23.2		14.5	20.3	22914.2	3
SPG 30 TERRACE	13.2	6.6	4.6	5.3		3.3	4.6	7853.5	
SPG 31 NORMAL	0.0	0.0	0.0	0.0		0.0	0.0	0.1	0
SPG 31 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	
SPG 32 NORMAL	1.5	0.9	0.5	0.7		0.5	0.6	54752.7	0
SPG 32 TERRACE	0.0	0.0	0.0	0.0		0.0	0.0	0.1	

```
STATEMENTS EXECUTED=  506581

CORE USAGE      OBJECT CODE=    3888 BYTES,ARRAY AREA=     344 BYTES,TOTAL AREA AVAILABLE=  114688  BYTES

DIAGNOSTICS       NUMBER OF ERRORS=       0, NUMBER OF WARNINGS=       0, NUMBER OF EXTENSIONS=      0

COMPILE TIME=     0.07 SEC,EXECUTION TIME=    25.56 SEC,     11.07.01    WEDNESDAY    24 MAY 78    WATFIV - JUN 1977 V1L6

        C/STOP
```